commodore Magazine

AÑO IV Núm. 38
Abril 1987 300 Ptas.

REVISTA INDEPENDIENTE PARA USUARIOS

Los secretos del octavo bit

Introducción al C

Juegos de ingenio: Las ocho damas y...

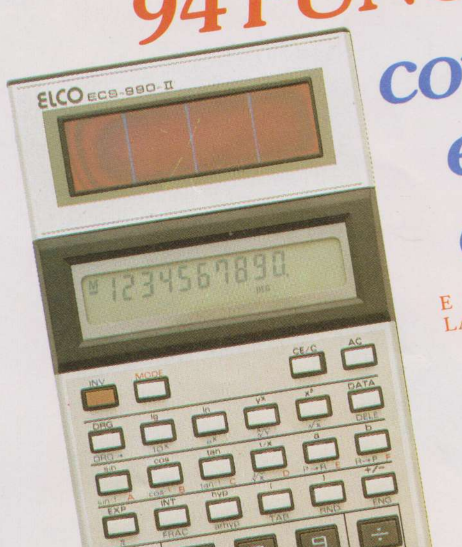
SUPLEMENTO
PROGRAMAS
12 PAGINAS

LAPICES OPTICOS

ELCO calculadoras para estudiantes:

94 FUNCIONES

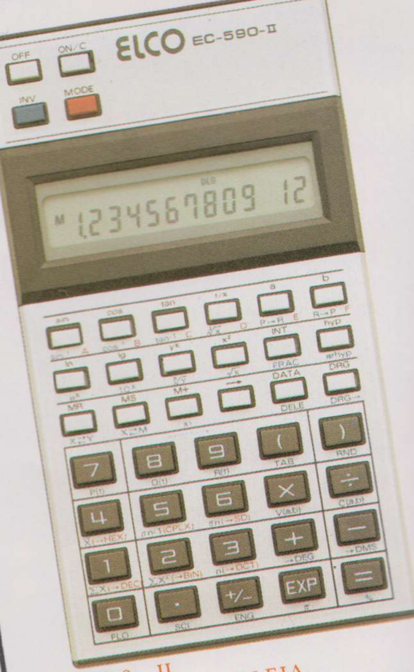
con cálculos y conversiones en decimal, hexadecimal, octal y binario. 4.590.-



E C S - 9 9 0 I I
LA CIENTIFICA SOLAR

Pantalla en LCD de 12 dígitos (10+2).
Funciones trigonométricas, exponenciales, logarítmicas, estadística e hiperbólicas y sus inversas.
Conversiones de grados centesimales a sexagesimales y de coordenadas rectangulares a polares y viceversa.
15 niveles de paréntesis.
Notaciones científicas, ingenieril o con selector de décimales.
Celdas solares de alta resolución.

5.590.-



E C - 5 9 0 I I
LA CIENTIFICA COMPLEJA

– Pantalla en LCD de 12 dígitos (10+2).
Funciones trigonométricas, exponenciales, logarítmicas, hiperbólicas y sus inversas.
Conversiones de grados centesimales a sexagesimales de coordenadas rectangulares a polares.
Funciones estadísticas: N, x, x^2, s, σ, DATA, CD, CAD, %.
Notaciones científicas, ingenieril o con el número de decimales deseado en pantalla.

E C - 1 0 0 P N
LA ECONOMICA
31 funciones con estadística y 8 dígitos.
Usa dos pilas normales.
2.990 ptas.



E C - 3 9 0 L A L I G E R A
31 Funciones con estadísticas y 8 dígitos.
Apagado automático.
3.290 ptas.



E C P - 3 . 9 0 0
LA PROGRAMABLE
Admite dos programas y 45 pasos de programación en memoria constante.
Con toma de decisiones.
64 funciones científicas y 10 dígitos.
6.590 ptas.

ALVARO SOBRINO



ELCO-1 Electrónica de Consumo-1, S.A.
c/ Rufino González, 6
Telfs.: 204 76 56 y 204 05 70 - Telex 42489 ELCO E
28037 MADRID

Director:
Rubén Sanz
Colaboradores:
José D. Arias
Alejandro de Mora-Losana
Paloma Saco
Diseño:
Benito Gil
Montaje:
Angel Martínez
Edita
PUBLINFORMATICA
Bravo Murillo, 377 - 5.º A
Telf.: 733 74 13. Madrid - 28020
Presidente:
Fernando Bolín
Director Editorial
Revistas Usuarios:
Juan Arencibia
Director de ventas:
Antonio González
Jefe de Producción:
Miguel Onieva
Servicio al cliente:
Julia González - Telf.: 733 79 69
Coordinadora Publicidad:
Silvia Bolín
Publicidad Madrid:
Emilio García
Publicidad Barcelona:
María del Carmen Ríos
Pelayo, 12
Telf.: (93) 301 47 00
ext. 27-28 y (93) 318 02 89
08001 BARCELONA
Depósito Legal: M-6622-1984
Dirección, Redacción y Publicidad:
Bravo Murillo, 377 - 5.º A
Teléf.: 733 74 13
Distribuye: S.G.E.L.
Avda. Valdeparra, s/n.
Alcobendas. Madrid
Distribuidor en Venezuela:
SIPAM, S. A.
Avda. República Dominicana
Edif. FELTREE
Boleita Sur Caracas (Venezuela)
Distribuidora en Argentina:
Distribuidora Intercontinental
Sta. Magdalena, n.º 541
Buenos Aires (Argentina)

Fotocomposición: Consulgraf
Nicolás Morales, 34. 28019 Madrid
Fotomecánica: Karmat
Pantoja, 10. Madrid
Imprime: G. Velasco, S. A.
Esta publicación es miembro
de la Asociación de Revistas
de la Información, asociada
a la Federación Internacional
de Prensa Periódica FIPP.
El P.V.P. para Ceuta, Melilla y
Canarias, incluido servicio
aéreo, es de 300 ptas. sin IVA

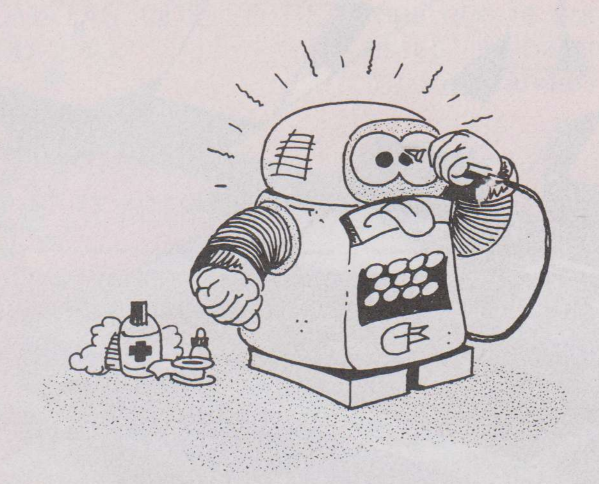
Rogamos dirijan toda la correspondencia relacionada con suscripciones a Commodore Magazine. Edisa. Tel.: 415 97 12. López de Hoyo's, 141-5. 28002 Madrid. Para todos los pagos reseñar solamente Commodore Magazine.
Para la compra de ejemplares atrasados dirigirse a la propia editorial Commodore Magazine, Bravo Murillo, 377 5º A. Tel.: 733 74 13. 28020 Madrid.

Editorial

Continuamos en este número con el estudio de otro de los periféricos disponibles para tu Commodore, el Lápiz Optico, que intento encontrar salida en el diseño por ordenador, pero que no ha encontrado la aceptación que se espera. Finalizando esta serie de artículos el mes que viene con las Tabletas Gráficas.

Una introducciòn al C te puede ayudar a entender la versatilidad de los nuevos y potentes lenguajes que permiten trabajar a nivel de bit, casi como el código máquina o simplemente trabajar a un nivel alto de programación, limitándonos a la utilización de las librerías o rutinas existentes para el lenguaje C.

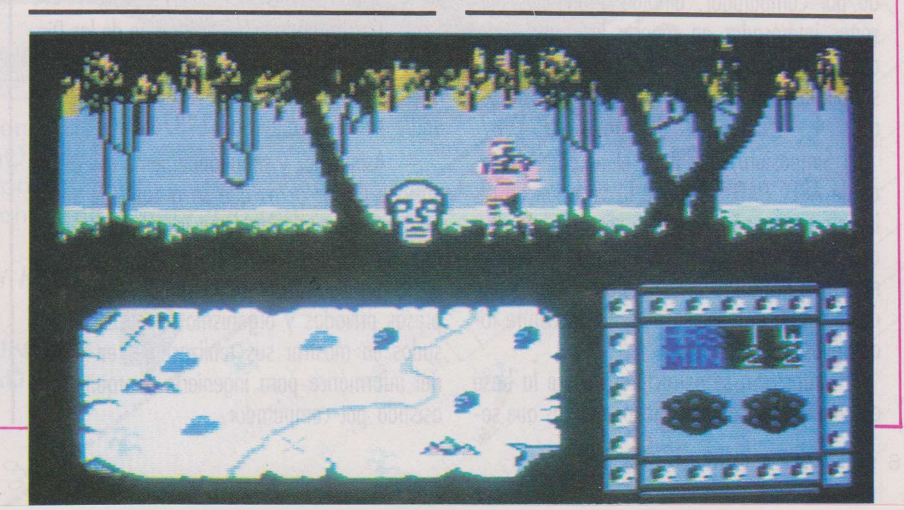
Damos importancia a este lenguaje, pues te permite una programación más estructurada; que a nosotros, acostumbrados al BASIC, lenguaje desordenado por naturaleza y que nos impide, entre otras, y como podréis ver en los juegos de ingenio la programación recursiva. Puede resultarnos muy útil comenzar nuestras andadas por un lenguaje que no dudamos que en poco tiempo acapare, respaldado por el sistema operativo UNIX, el campo de programación en aplicaciones tanto empresariales como de diseño gráfico.



Esta revista no mantiene relación de dependencia de ningún tipo con respecto de los fabricantes de ordenadores Commodore Business Machines ni de sus representantes.

Suma

rio



NOTICIAS



Universidad Politécnica de Madrid
Escuela Técnica Superior
de
Ingenieros de Caminos, Canales y Puertos

TECNICAE-87

Del 4 al 8 de mayo de 1987, va a tener lugar en los locales de la Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos de Madrid, «TECNICAE-87», I Semana de la Ingeniería Asistida por Computador, que va a llevarse a cabo bajo el patrocinio de los Ministerios de Educación y Ciencia e Industria y Energía, la Comunidad Autónoma de Madrid y el Rectorado de la Universidad Politécnica de Madrid.

El objetivo de estas Jornadas es llevar al ámbito universitario una visión lo más completa posible del estado actual y de las posibilidades de utilización que ofrecen los sistemas de diseño por computador en el ámbito tecnológico, y al tiempo, servir de punto de encuentro entre los mundos de la Universidad y la Empresa.

A lo largo de los cinco días de duración de «TECNICAE-87» tendrán lugar, simultáneamente, las siguientes actividades.

— Un Curso de Iniciación al Diseño Asistido por Computador, dirigido a aquellas personas interesadas en conocer los principios y alcance de los sistemas CAD/CAM, que se desarrollará a lo largo de las mañanas de los días 4, 5, 6 y 7 de mayo de 1987.

El Simposium, que lleva el nombre genérico de AREC.DAO'87, tiene la vocación de celebrarse cada año, y quiere ser una iniciativa que reúna arquitectos, ingenieros, diseñadores, informáticos e investigadores, con objeto de fomentar el indispensable diálogo entre todos ellos.

El Simposium se estructurará sobre la base de ponencias o comunicaciones escritas que serán presentadas a los asistentes, así como publicadas.

ARED.DAO '87 pretende especialmente impulsar los siguientes temas:

1. Proceso de diseño en la arquitectura y en la construcción: metodología.
2. Bases de datos alfanuméricos y gráficos.
3. Representación gráfica: algoritmos, normalización , etcétera.
4. Sistemas integrados.

Para cualquier consulta, ampliación de la información o inscripciónes, los interesados pueden dirigirse al ITEC, Instituto de Tecnología de la Construcción de Cataluña, C/ Bon Pastor, 5, 4.º. Tel.: 209 60 99. 08021 Barcelona (España).

— Una serie de Jornadas Sectoriales sobre Ingeniería Asistida por Computador (CAE), que se celebrarán durante las tardes de los días 4, 5, 6 y 7 de mayo de 1987, en sesiones paralelas.

— Una Jornada sobre la Función de la Universidad en la implantación y transmisión de las tecnologías CAE, que tendrá lugar a lo largo de la mañana del día 8 de mayo de 1987.

— Una Jornada sobre la Función de las Distintas Administraciones en la Transmisión de las Tecnologías CAE, que se celebrará en la tarde del 8 de mayo de 1987.

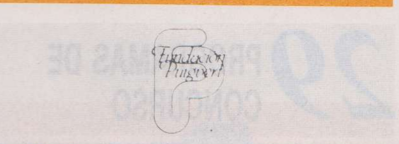
— Asimismo, y simultáneamente con el desarrollo de los actos que se acaban de señalar, se celebrará una Exposición que se acaban de señalar, se celebrará una Exposición en la que estarán presentes una serie de empresas privadas y organismos públicos interesados en mostrar sus realizaciones en material informático para ingeniería y arquitectura asistida por computador.

El «Supermicros 86» para la impresora Láser Canon LBP-8

El pasado día 26 de febrero, en el hotel Eurobuilding de Madrid, la revista Micros hizo la entrega a los representantes de Canon España, S. A., de su trofeo anual «Supermicros», concedido en su versión 86 a la impresora Láser Canon LBP-8, en la categoría de «periferia de impresión».

Con este modelo, Canon se interna en el campo de la tecnología láser aplicada a la informática, en el cual Canon es el líder indiscutible.

Este premio es el reconocimiento de la prensa especializada a la constante preocupación de Canon por abrir nuevos caminos en el mercado español, dando entrada a los modelos más avanzados a nivel mundial. Una vez más... Enhorabuena, Canon.



2.ª Jornada de Informática en los Servicios de Anestesia y Reanimación

Día: 20 de junio de 1987.
Lugar: Aula Serés.
Fundación Puigvert.
C/ Cartagena, 340.
08025 Barcelona.
Telf. 235 56 14.

Precio de la inscripción: 5.000 ptas. (con. almuerzo incluido).

Organizado por la Fundación Puigvert en colaboración con la Sección de Informática de la Sociedad Española de Anestesiología y Reanimación.

Caja de seguridad para discos

Advantek International Limited, de Hong Kong, ha lanzado al mercado una caja fuerte, con cerradura y a prueba de incendio, llamada *«Disksafe».* Está hecha con un metal que, además de proteger su contenido del fuego, reduce el riesgo de que el disco o cinta queden perjudicados por cualquier campo magnético presente.

Su diseño es bastante atractivo y no ocupa demasiado espacio, puesto que mide 210×190×350 cm. Su capacidad es de 80 discos de cualquier tamaño y aún queda espacio libre para cintas.

Otra novedad de esta empresa es el ordenador para los deportistas.

El *Sports Computer* es un aparato moderno, diseñado para las personas que quieren ponerse en forma. Se trata de una máquina muy sencilla que mide el ritmo de velocidad, proporciona un nuevo aliciente al deporte y vigila la salud del usuario controlando su pulso. Se han diseñado varias versiones del *Sports Computer,* para varios deportes específicos, tales como bicicleta fija, remo, jogging y esquí y puede adaptarse a cualquier tipo de actividad. Hay varios modelos de *Sports Computer* en el mercado, poseyendo esencialmente las mismas características. Para obtener más información, dirigirse a:

Hong Kong Trade Developement Council
Balmes, 184.
08006 Barcelona.

ARECDAO 87

ARECDAO '87

I Simposium Internacional sobre Diseño Asistido por Ordenador en Arquitectura e Ingeniería Civil.

1-3 de abril de 1987, Barcelona (España).

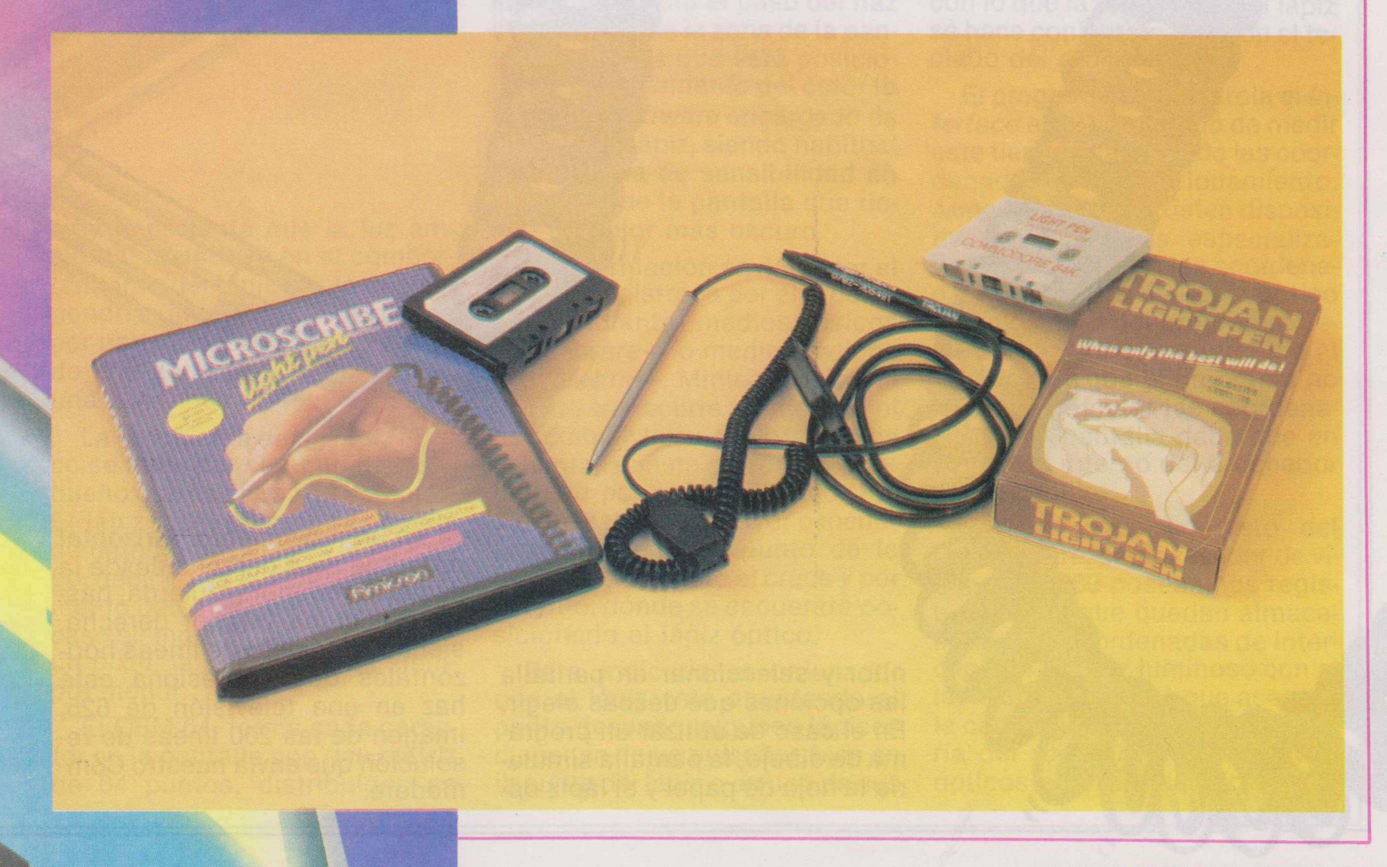
Organizado por el ITEC, Instituto de Tecnología de la Construcción de Cataluña y patrocinado por el MOPU-Ministerio de Obras Públicas y urbanismo, en el marco de CONSTRUMAT-87, se celebrará en Barcelona del día 1 al 3 de abril de 1987, AREC.DAO'87, I Simposium Internacional sobre Diseño Asistido por Ordenador en la Arquitectura e Ingeniería Civil. La inauguración será a cargo del Excmo. Sr. Don Javier Luis Sanz Cosculluela, ministro de Obras Públicas y Urbanismo.



ESTE LAPIZ ME SERVIRA
HAY QUE MODERNIZAR LA PROPIA IMAGEN
¡UN ORDENADOR JAPONES!
SINTES

Lápices ópticos

Entre los Periféricos conectables a los *port* de juego, podemos encontrar Joysticks, Ratones, Tabletas gráficas y Lápices ópticos entre otros. Estos últimos van a ser objeto de estudio en este artículo para que puedas llegar a conocerlos y comprenderlos mejor.



El lápiz óptico es un periférico que nos permite introducir datos en el ordenador. El *hardware* consiste en un dispositivo con forma de lapicero que está unido al ordenador por un cable conectado al *port* de juego.

La manera de utilizar este lapicero depende del *software* que le acompañe, pero podemos decir que la mecánica de uso está en acercar éste al televisor o monitor y seleccionar en pantalla las opciones que deseas elegir. En el caso de utilizar un programa de dibujo, la pantalla simularía la hoja de papel y el lápiz óptico actuaría como el elemento que dibuja.

La aplicación más usual de los lapiceros ópticos es en los programas de proceso de gráficos, donde resulta más simple apuntar mediante el lápiz óptico a una zona de la pantalla donde queremos actuar, a esperar que llegue el cursor activado por un *joystick* o un ratón.

FUNCIONAMIENTO

El funcionamiento de un lápiz óptico es muy simple:

Conozcamos primero cómo aparecen las imágenes en la pantalla de un televisor, para luego poder analizar el funcionamiento de los lapiceros ópticos.

La imagen en pantalla del monitor se forma a partir del barrido de un haz de electrones que impresiona la capa fluorescente depositada en el interior del tubo de pantalla. El haz se desplaza en un movimiento horizontal de izquierda a derecha desde la esquina superior izquierda, hasta la esquina inferior derecha, siendo el número de líneas horizontales que impresiona este haz en una televisión de 625, imagen de las 200 líneas de resolución que envía nuestro Commodore.

El lápiz óptico se convierte en un dispositivo capaz de detectar el paso del haz electrónico, con el fin de averiguar el punto de pantalla en el que se encuentra, cuando se produce el cruce con el haz. Para esto, el lápiz óptico está provisto de un elemento sensible a la luz, de tal manera que puede detectar un cambio brusco de intensidad luminosa, como lo es el paso del haz. Es muy importante que la luz ambiental no influya en el comportamiento del lapicero, lo que supondría unos disparos bruscos por parte de éste, al desplazarlo de una zona poco iluminada a una zona más iluminada.

La resolución de un lápiz óptico se mide por el punto más pequeño que es capaz de detectar. El lápiz ideal debería ser capaz de detectar un *pixel,* sin embargo, debido a la superficie del elemento detector que en cualquier caso siempre es mayor que la de un *pixel,* la resolución es siempre menor.

El CBM-64, forma cada carácter en la pantalla con una matriz de 64 puntos, distribuidos en ocho filas por ocho columnas, sin embargo, habitualmente, el lápiz óptico divide la misma superficie en ocho bloques debido a su menor resolución.

Hay que añadir que el lápiz solamente detecta el paso del haz y no el color de la zona de la pantalla, sobre la que está posicionado. El tratamiento del color lo realiza el *software* encargado de manejar el lápiz, siendo habitual una pérdida de sensibilidad en las zonas de la pantalla que tienen un color más oscuro.

La información enviada por el lápiz es registrada por el ordenador mediante medios físicos —*hardware*— o medios lógicos —*software*—. Midiendo el tiempo que transcurre desde que el haz comienza su exploración de la pantalla hasta que éste, es detectado por el lápiz óptico. Pudiendo saber con mayor o menor precisión, en qué punto de la pantalla se detectó el cruce y por lo tanto, dónde se encuentra posicionado el lápiz óptico.

Para indicarle al ordenador que el lápiz está apuntando al punto deseado, algunos lápices disponen de un pulsador, bien en la punta del lápiz o en uno de sus lados, encontrando también en algunos modelos pulsadores del tipo táctil en los que sencillamente con tocar la zona sensible se obtiene el mismo efecto que con un pulsador normal. Otros modelos, sin embargo, carecen de cualquier tipo de pulsador, con lo que la utilización del lápiz se hace conjuntamente con el teclado del ordenador.

El programa que controla el *interface* es el encargado de medir este tiempo, calculando las coordenadas de posicionamiento. Aunque también existen dispositivos electrónicos especializados que contienen las coordenadas del disparo del lápiz óptico a disposición del programa. Este segundo tipo es preferible al primero, ya que el programa no necesita calcular las coordenadas de la pantalla dado que un elemento interno del ordenador se las facilita.

Para el caso concreto del Commodore, el procesador de video VIC 6569 posee unos registros en los que quedan almacenadas las coordenadas de intercepción del haz luminoso con el lápiz óptico. Hecho que asegura la compatibilidad entre la mayoría del *software* de los lápices ópticos en el Commodore.

EL LAPIZ OPTICO TROJAN

El paquete que nos ofrece la casa Trojan, incluye el lápiz óptico además de un manual que te permite utilizar el *software,* presentado en *cassette.*

El lápiz Trojan, de origen inglés, fue uno de los pioneros dentro del mercado de diseño gráfico por ordenador.

Hardware:

Tras un conector de nueve *pines* y un cable de un metro, encontramos el lápiz, propiamente dicho. Este se encuentra formado por una funda de plástico, en cuyo interior encontramos algunos componentes electrónicos, entre ellos un fototransistor, una resistencia y un transistor bipolar convencional.

La conexión entre la parte electrónica y el *port* de juego se realiza sencillamente a través de tres cables, dos de alimentación y uno portador de la señal de disparo del periférico.

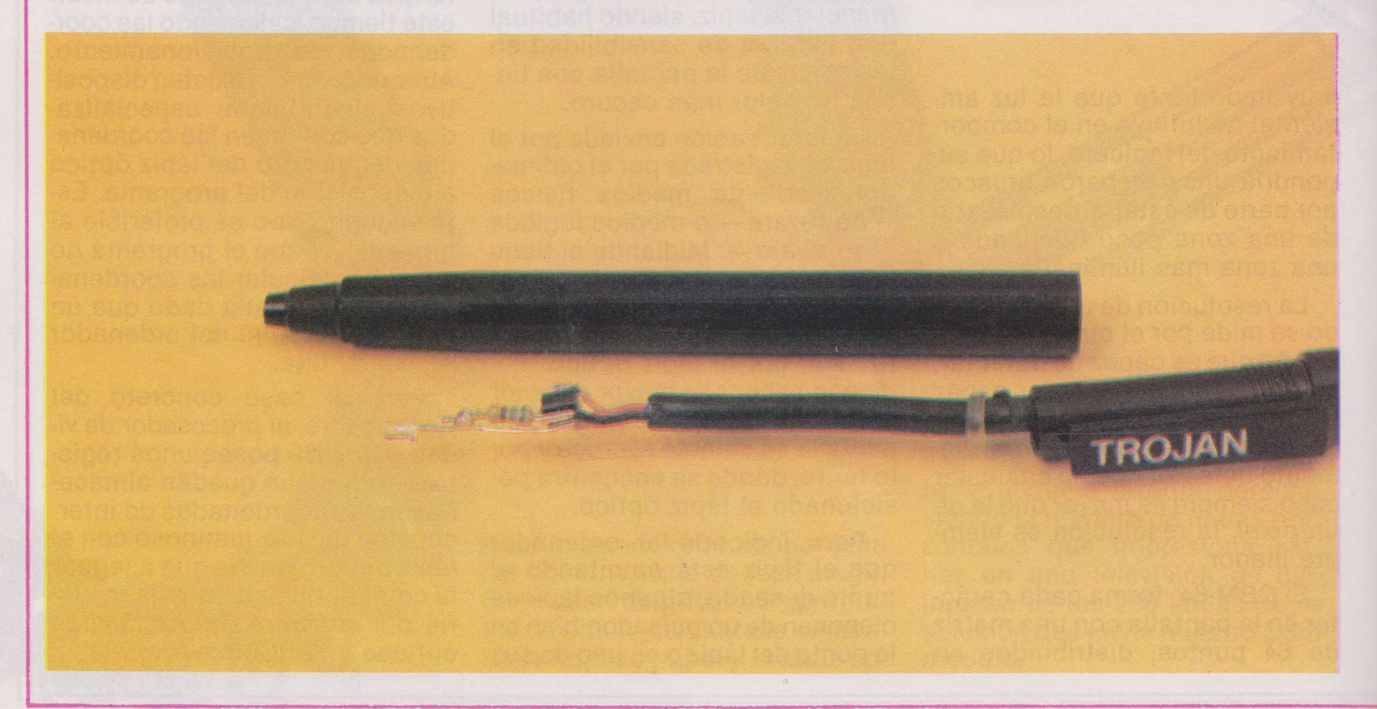
El índice de fiabilidad de este lápiz óptico es muy alto, debido a la poca cantidad de componentes necesarios para cumplir su función. Físicamente tiene un diseño que facilita su manejo con suma comodidad, además de presentar una gran robusted.

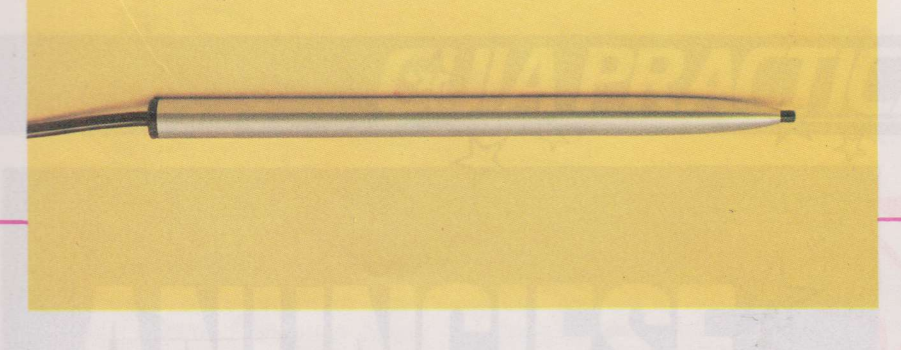
Software:

El programa de diseño gráfico CADMASTER que acompaña al lápiz, en forma de *cassette,* aunque también soporta disco, tiene todas las características de un buen programa de proceso de gráficos; el menú principal ocupa toda la pantalla en forma de pequeños cuadrados, sobre los que se realiza la selección y un texto en cada cuadrado que indica la acción que cada opción realiza.

En conjunto se trata de un paquete con una buena relación calidad-precio.







EL LAPIZ OPTICO MICROSCRIBE

El distribuidor en España de la casa Amicron, Enfa Ibérica, nos ofrece el lápiz óptico Microscribe a un precio de 10.000 pesetas.



Hardware:

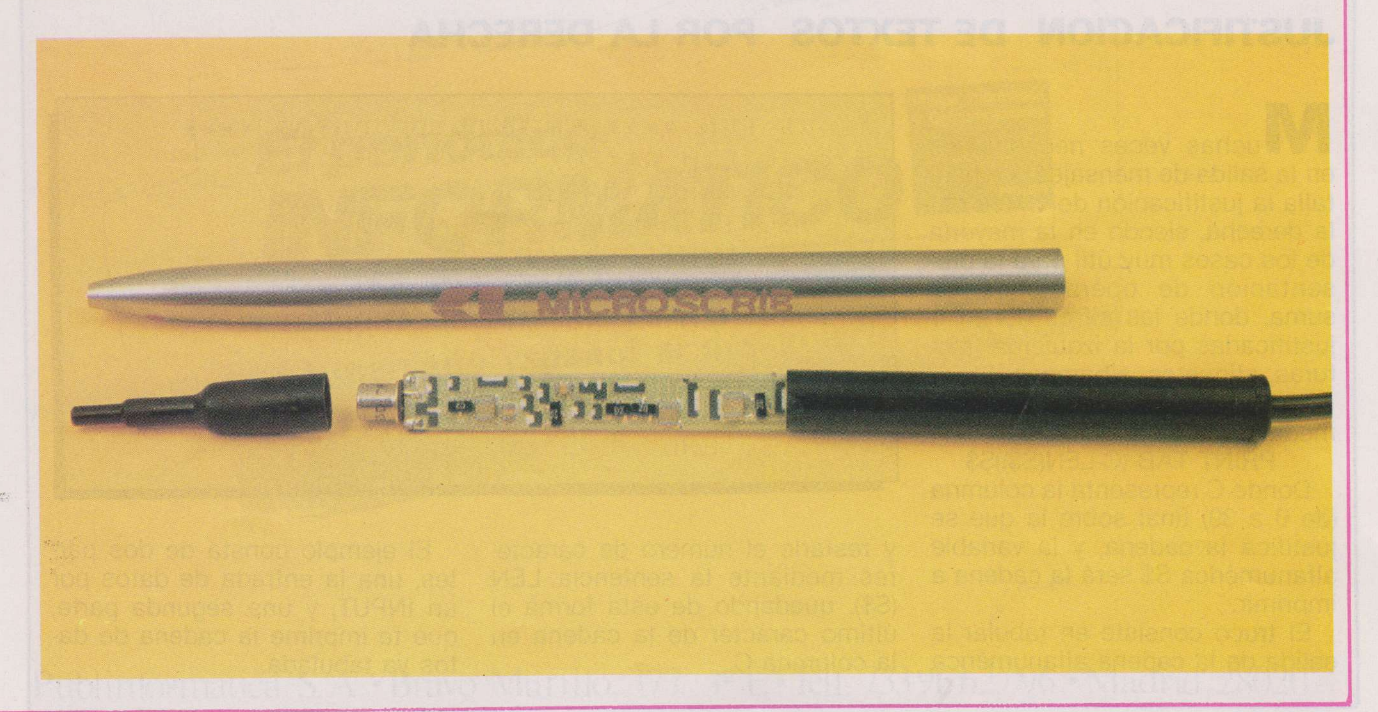
La parte electrónica de este lápiz, presenta un diseño y características totalmente profesionales. El «envase» de este lápiz, con un diseño muy estilizado, es de metal y dejando asomar al exterior por la parte anterior un pulsador. En el interior del tubo metálico que sirve de protección física al circuito además de protegerle contra las radiaciones electromagnéticas, encontramos una placa-soporte de un material cerámico, en la que están serigrafiadas las resistencias del circuito. El proceso de ajuste de estas resistencias se realiza a través de unas incisiones realizadas con un rayo láser, hasta alcanzar el valor ópmico necesario en cada una de ellas. El resto de los componentes, en su mayoría transistores de un diminuto tamaño, están soldados en la superficie del sustrato cerámico. Este tipo de tecnología es poco habitual en la electrónica de consumo y menos habitual en periféricos de bajo precio, por lo que la calidad de este lápiz es excepcional.

Software:

En una *cassette* con turbo, encontramos el programa de diseño gráfico llamado GRAFFITI y en la documentación que acompaña al lápiz encontramos las rutinas para manejar este periférico desde nuestros propios programas BASIC (muy de agredecer). El menú principal, siempre presente en la pantalla junto con la zona de dibujo, está formado por unas ventanas en la parte inferior de la pantalla, con un texto que indica la acción que realizan. Algunas de las opciones hacen aparecer un nuevo menú de selección. El disco también se encuentra soportado en este paquete.

En resumen, un lapicero técnicamente perfecto junto con un *software* aceptable.

Enrique Fernández Rasero





CENTRADO DE TEXTOS

Ahora con tu Commodore y con la ayuda de este truco puedes sacar textos centrados en tu pantalla; sólo tienes que introducir una pequeña rutina de centrado y utilizar un PRINT TAB (FN C (C)) A$ para la impresión del texto.

La rutina de centrado determina la columna de pantalla donde va a comenzar la impresión de la variable A$, en el caso del ejemplo es la variable A$(P), en lo que nosotros hemos llamado rutina de impresión, se pinta el valor de la variable alfanumérica A$(P) con la lontigud determinada por LEN(A$(P)) y en la posición calculada por FN C(C).

```
9 REM **** ENTRADA DE DATOS ****
10 DIM A$(20)
20 INPUT"DATOS":A$
30 IF A$="FIN" THEN 100
40 N=N+1 :A$(N)=A$
50 GOTO 20
99 REM ** RUTINA DE IMPRESION **
100 FOR P=1 TO N
110 GOSUB 200
120 PRINT TAB(FN C(C))A$(P)
130 NEXT P
140 END
199 REM ** RUTINA DE CENTRADO **
200 C=LEN(A$(P))
210 DEF FN C(C)=20-C/2
220 RETURN
```

Todas estas pequeñas rutinas puedes introducirlas en cualquiera de tus programas, estudiando primero los parámetros que tienes que ordenar.

JUSTIFICACION DE TEXTOS POR LA DERECHA

Muchas veces necesitamos en la salida de mensajes por pantalla la justificación de textos por la derecha, siendo en la mayoría de los casos muy útil para la presentación de operaciones de suma, donde las cifras deben ir justificadas por la izquierda (facturas, etiquetas, albaranes...).

La forma de conseguir esto es mediante la instrucción.

PRINT TAB (C-LEN(S$))S$

Donde C representa la columna (de 0 a 39) final sobre la que se justifica la cadena, y la variable alfanumérica S$ será la cadena a imprimir.

El truco consiste en tabular la salida de la cadena alfanumérica y restarle el número de caracteres mediante la sentencia LEN (S$), quedando de esta forma el último carácter de la cadena en la columna C.

```
9 REM **** ENTRADA DE DATOS ****
10 DIM A$(20)
20 INPUT"DATOS":A$
30 IF A$="FIN" THEN 100
40 N=N+1 :A$(N)=A$
50 GOTO 10
99 REM ** RUTINA DE IMPRESION **
100 FOR C=1 TO N
110 PRINT TAB(20-LEN(A$(C)))A$(C)
120 NEXT C
130 END
```

El ejemplo consta de dos partes, una la entrada de datos por un INPUT, y una segunda parte, que te imprime la cadena de datos ya tabulada.

GUIA PRACTICA

ANUNCIESE por MODULOS

LOBERCIO
COMPUTER - CENTER

Unico en España:

- DISPONEMOS TODO TIPO DE REPUESTOS Y MANUALES DE REPARACION DE COMMODORE
- REPARACION RAPIDA Y GARANTIZADA
- TODA CLASE DE PERIFERICOS 64/128
- TONER PARA COPIADORA CANON- MINOLTA

Consultas: Tel.: (952) 33 27 26
Avda. de Andalucia, 17 29002 MALAGA

MADRID
(91) 733 96 62
BARCELONA
(93) 301 47 00

La industria informática española tiene lo que necesita.


El periódico
INFORMATICO

Philips Informática y Comunicaciones
TECNOLOGIA Y SERVICIO
PHILIPS

Fabricantes y proveedores hardware y software

El mercado español al descubierto

Nixdorf espera facturar unos 15.000 millones este año

Publinformática. S.A. • Bravo Murillo. 377. 3º E • Telf. 733 96 62 / 96 • Madrid 28020



Speed-King

Seguro que a estas alturas ya eres un experto jugador de Speed-King; y digo esto porque el juego del que hablamos ahora ya está presente en muchos locales públicos. Me estoy refiriendo a una peligrosa carrera de motos.

Puedes elegir entre diez circuitos, con distintas dificultades en cada uno. Antes de comenzar la carrera tienes la oportunidad de dar una vuelta de reconocimiento por el circuito elegido. Como ya sabemos, tu objetivo es participar en una gran carrera de motos. Competirás con otros diecinueve corredores. Si tienes poca práctica en este tipo de carreras, no te preocupes, porque tienes la oportunidad de entrenar sobre el circuito las veces que quieras, antes de comenzar la competición. Comienza por el nivel de «novicio», continúa con «prácticas» y luego lánzate a la carrera.

Speed-King tiene un elemento que da una emoción adicional al juego. Tu moto está especialmente preparada para grandes competiciones y tiene hasta seis marchas, que ajustarás dependiendo de la velocidad a la que circules.

Durante el recorrido, podrás estar informado de todas las incidencias que se produzcan. En la parte superior de la pantalla, verás unos indicadores que te informan sobre la velocidad que llevas, el número de vueltas que has dado al circuito y el mejor tiempo conseguido hasta el momento. También dispones de un cronómetro, al que debes estar atento en todo momento, para mejorar tu récord de tiempo.

Si tienes mala suerte (o poca práctica) y chocas con algún otro participante en la carrera, perderás combustible y sufrirás una penalización de tiempo, hasta que consigas volver a tener control sobre la moto.

Para obtener un mejor resultado en la competición, puedes reducir el número de vueltas que vayas a dar al circuito. Pero ten en cuenta que si tomas esta medida, tu habilidad en el manejo de la máquina tendrá que ser mucho mayor.



FICHA DEL JUEGO

NOMBRE:	**SPEED-KING**
DE:	**MASTERTRONIC (ABC Soft)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK O TECLADO**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD						■				
ADICCION								■		
GRAFICOS							■			
SONIDO							■			
GENERAL							■			





War Hawk

FICHA DEL JUEGO
NOMBRE: **WAR HAWK**
DE: **FIREBIRD (Drosoft)**
ORDENADOR: **COMMODORE-64, 128**
CONTROL: **JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD					■					
ADICCION									■	
GRAFICOS								■		
SONIDO								■		
GENERAL							■			

Estamos esta vez, ante un juego que tiene todo el encanto de las antiguas máquinas de «marcianitos», junto con la sofisticación en gráficos y sonidos característica de los juegos de ordenador de hoy en día.

Si te gustan los juegos en los que hay que disparar continuamente contra las naves enemigas, y esquivarlas con toda rapidez, estás de enhorabuena, ¡ya existe tu juego favorito!

War Hawk, es una nave espacial, de avanzada tecnología que tú pilotas por la zona del Asteroide. Pero, como podrás imaginar, no se trata de un viaje de placer, sino que tendrás que ir defendiéndote constantemente de tus enemigos. Ellos procurarán no dejarte en paz, enviando contra ti, las más extrañas naves que puedas imaginar. Además, de vez en cuando, se interpondrán en tu camino unas enormes minas. Procura no chocar con ellas, o de lo contrario, tus escudos protectores irán desapareciendo y quedarás indefenso ante tantos enemigos.

Es fundamental que durante todo el recorrido, seas capaz de conservar la calma. No dejes que la velocidad del juego o los sonidos, tan realistas en ocasiones, puedan afectar a tus nervios. Tu mano debe ser segura, y no vacilar en ningún movimiento si quieres formar parte del *ranking* de héroes del espacio.

Aunque el argumento y el desarrollo del juego que tratamos, son frecuentes en juegos de ordenador, Warhawk destaca por la alta calidad que sus creadores han sabido dar tanto a los gráficos como al sonido. Tiene todos los ingredientes para crear un alto nivel de adicción tanto en personas muy aficionadas a los juegos de ordenador, como en los principiantes o incluso en aquellos a los que no les gustan este tipo de juegos.

Cuando pienses jugar con Warhawk, asegúrate de que tienes mucho tiempo por delante; nunca se sabe, ¡cuándo te decidirás a desenchufar el ordenador!

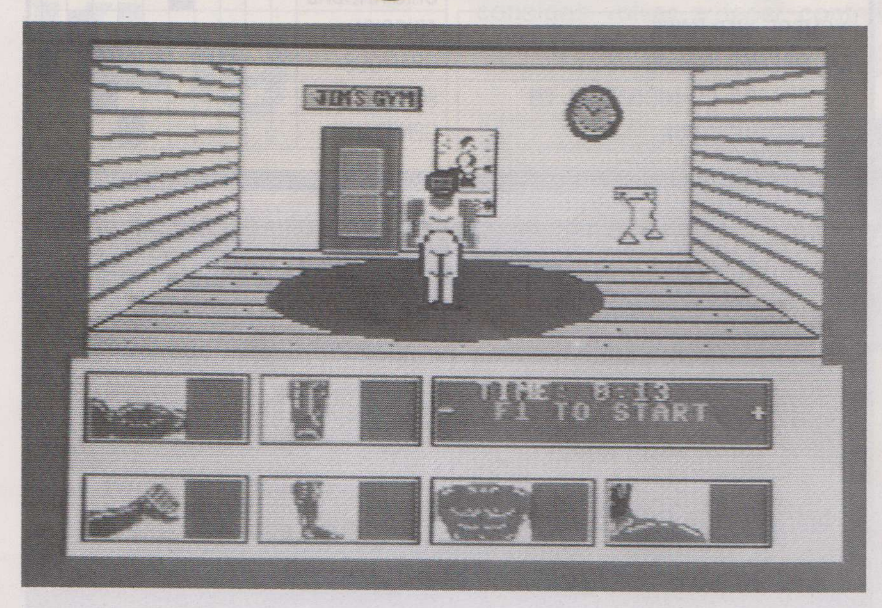








Strongman



Ultimamente está muy de moda eso de desarrollar los músculos, cultivar el cuerpo, el aerobic, culturismo, etc... está siendo algo común para muchas personas. Pues bien, el juego de Zafiro Soft, te dará unas cuantas ideas para cultivar los músculos.

Nuestro protagonista es GEOFF CAPES, campeón de los Juegos Mundiales de Montaña, y con él te entrenarás para poder competir contra los más difíciles adversarios.

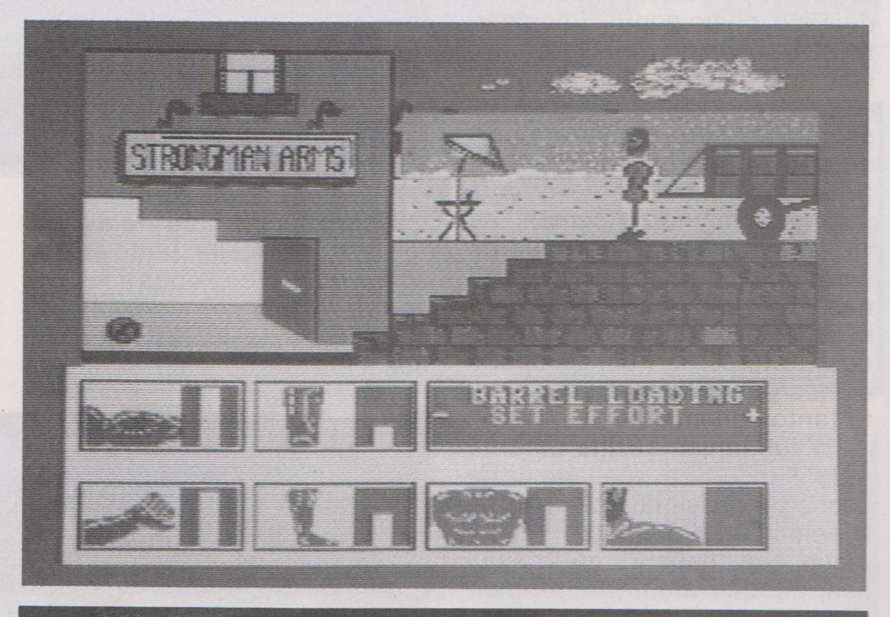
El juego tiene dos partes, una el entrenamiento y otra la competición. En la parte del entrenamiento, deberás distribuir tu tiempo y los ejercicios que quieres hacer para fortalecer unas u otras zonas de tu cuerpo, para ello podrás decidir tú en que partes quieres tener más fuerza, o en un brazo, o en las piernas, etc... Los entrenamientos se hacen con pesas, o con barras de hierro que deberás doblar.

Cuando consideres que ya estás suficientemente preparado, podrás pasar a la acción, tienes seis posibles adversarios que son de lo más variados, desde arrastrar camiones, volcar coches, levantar toneles, cortar leños, etc... hasta la lucha libre.

Para logra pasar todos los niveles, tendrás que practicar mucho pues no es nada fácil, pero insiste, y a base de mucho entrenamiento conseguirás la fuerza necesaria para salvar todos los obstáculos.

Este juego presenta unos gráficos donde nos muestra el interior de un gimnasio con sus aparatos y, ya fuera de él, los duros retos que tendrá que superar para ser el Campeón de la Olimpiada de este año.

En general, es un juego de tema original, pero un poco lento, aunque si te gusta esta trama, ten por seguro que te hará estar mucho tiempo «entrenando».



FICHA DEL JUEGO

NOMBRE: **STRONGMAN**
DE: **MARTECH (Zafiro)**
ORDENADOR: **COMMODORE-64**
CONTROL: **JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD							■			
ADICCION					■					
GRAFICOS							■			
SONIDO						■				
GENERAL							■			





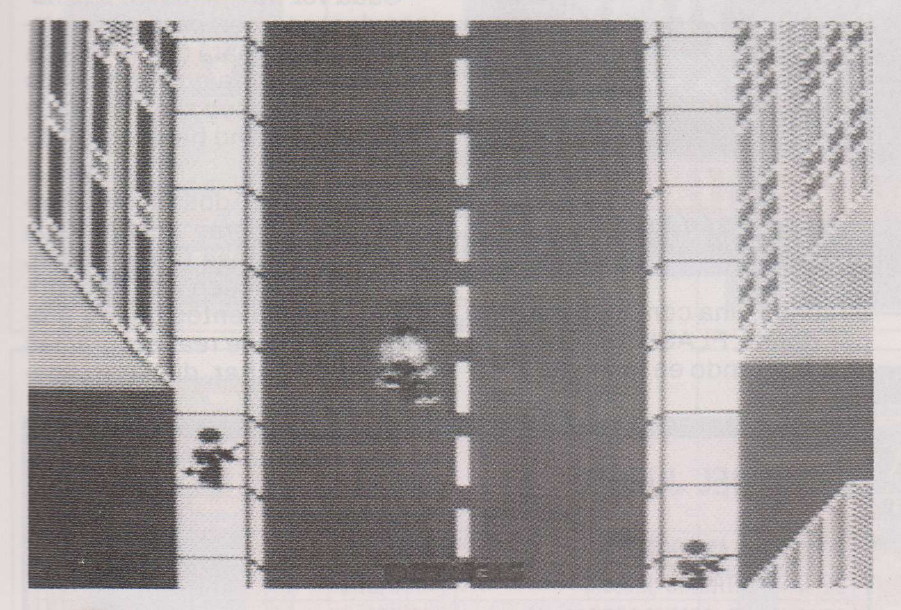


Swat

FICHA DEL JUEGO

NOMBRE:	**SWAT**
DE:	**MASTERTRONIC (Drosoft)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD						■				
ADICCION						■				
GRAFICOS							■			
SONIDO						■				
GENERAL						■				



¿Recuerdas aquella serie de T.V.? Sí, la que trataba de las misiones de un cuerpo de policía especialmente entrenado para misiones difíciles y peligrosas. Cuando ya casi se creía imposible ese «final feliz», llegaban los hombres de SWAT (los hombres de Harrelson) y lo solucionaban todo.

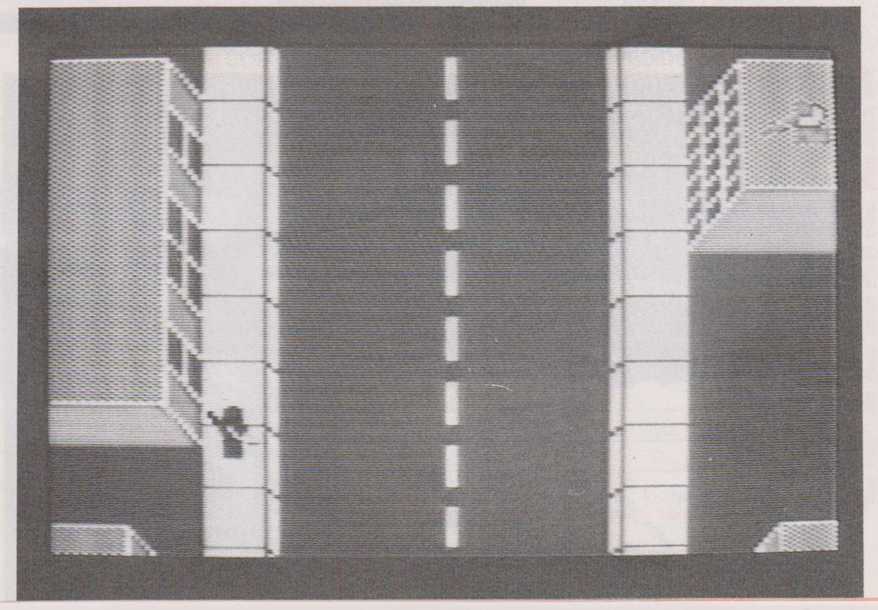
Bueno, pues una vez más, los SWAT se encuentran ante una misión difícil y muy importante. Nos encontramos en la ciudad de Los Angeles, y un grupo de terroristas ha conseguido hacerse fuerte en la parte oeste de la ciudad. Sólo los hombres de SWAT, tienen la suficiente pericia y valentía como para enfrentarse a esta situación y salir victoriosos.

La misión se complica especialmente al desarrollarse en las calles de la ciudad. Un auténtico ejército de hombres sin piedad, armados con bombas y metralletas, intentarán acabar con todo el equipo SWAT. No sólo habrá que acabar con los terroristas, además hay que proteger a la población.

Los enemigos son muy listos. Ellos pensaron que la policía no se atrevería a intervenir en plena ciudad, contra unos hombres armados hasta los dientes. Es cierto, la vida de muchos transeúntes inocentes corre peligro. Por ello, los SWAT deberán tener especial ciudado en afinar muy bien su puntería, y disparar sólo cuando sea necesario, y contra los terroristas. Si por error matas a un ciudadano inocente, serás gravemente sancionado. Un equipo fuerte y bien entrenado como SWAT no puede permitirse esa clase de fallos.

Del mismo modo, el enemigo intentará capturar rehenes que tú, como miembro del equipo SWAT, deberás rescatar sin que su vida corra peligro.

La calle está plagada de hombres armados. Distingue bien a tus enemigos de los inocentes. Dispara sin piedad contra los primeros y rescata y protege a los segundos, es tu misión ahora dentro del equipo SWAT.



Flash Gordon



Vuelve el gran héroe que no tenía rivales capaces de hacerle perder su fama, miles de aventuras enfrentándose a los mayores peligros jamás conocidos, hablamos de...FLASH GORDON.

Esta vez también le ha tocado una dura tarea, primero en la peligrosa jungla, luego peleando con un terrible enemigo... Tú tendrás que echarle una mano, o las dos, por esta vez.

La pantalla aparece dividida en zonas, una con un mapa, otra por donde FLASH GORDON se va adentrando en la jungla y luego un contador de tiempo cuyo límite son 24 horas.

En la selva te enfrentarás con terribles pájaros, precipicios interminables, fieras que te atacarán si no eres más rápido que ellas. Puedes acabar con tus enemigos usando tu arma, pero sólo la podrás recargar una vez en un sitio determinado donde dejaste tu munición, luego desaparece y no vuelve a aparecer, tu suerte dependerá de tu habilidad.

Cada vez que te maten o te hagan desaparecer, perderás tiempo extra y eso hará que tu misión se complique, ya que tienes que evitar los retrasos si quieres que todo acabe como habías planeado.

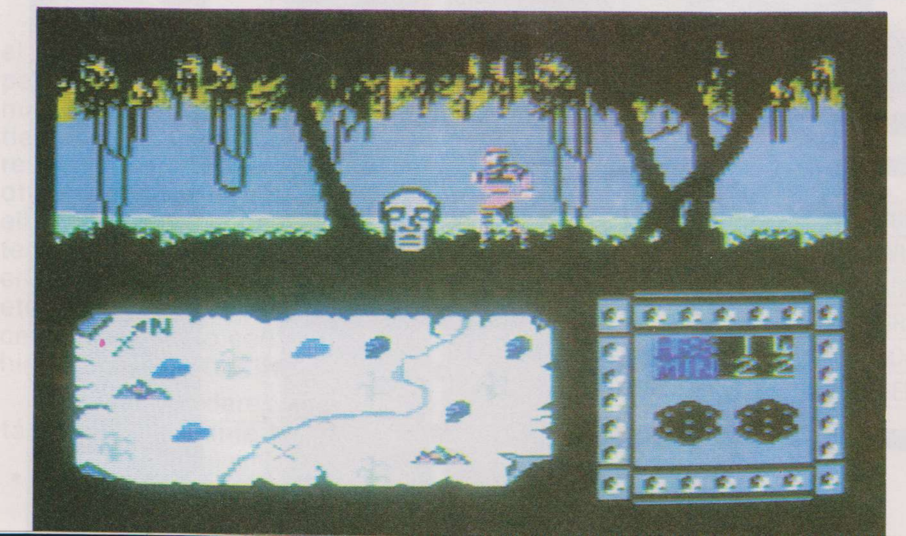
En los otros dos niveles tendrás que superar también los problemas que se te presenten. Una ayuda pueden ser, la cantidad de movimientos que tu protagonista puede realizar gracias al *joystick*: saltar, disparar, agacharse, pegar puñetazos, etc... Con un poco de paciencia lograrás dominar todos ellos, y así contribuir a que concluya su misión satisfactoriamente.

Los gráficos y el sonido adecuados para el tipo de juego de que se trata. El tema, bien elegido y tratado, es capaz de mantener la intriga durante bastante tiempo, así lo único que hace falta es que lo pongas en tu ordenador e intentes cumplir la misión que esta vez te ha tocado a ti resolver con una pequeña ayuda de FLASH GORDON.

FICHA DEL JUEGO

NOMBRE:	**FLASH GORDON**
DE:	**MASTERTRONIC (Drosoft)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD								■		
ADICCION							■			
GRAFICOS								■		
SONIDO								■		
GENERAL								■		





LE OFRECE LOS MEJORES LIBROS PARA SU ORDENADOR



P.V.P. 750 PTAS.
(IVA INCLUIDO)
Descubre los misterios de la programación de una forma sencilla, con ejemplos, programas y organigramas.
(110 páginas, tamaño 13,5 x 21)



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Con utilidades, juegos explosivos y gráficos dinámicos que lleva al BASIC hasta el mejor aprovechamiento de sus posibilidades.
(200 páginas, tamaño 15,5 x 21,5).



P.V.P. 750 PTAS.
(IVA INCLUIDO)
Un libro especialmente dedicado a los que se inician por vez primera en el mundo del Spectrum.
(100 páginas, tamaño 13,5 x 21).



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Una inestimable ayuda que complementará la que proporciona el manual del ordenador.
(108 páginas tamaño 13,5 x 21,5).



P.V.P. 900 PTAS.
(IVA INCLUIDO)
Un compendio de los programas más diversos con los que podrá aprender jugando las importantnes características del BASIC.
(258 páginas, tamaño 15,5 x 21,5).



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Muestra una visión más completa del correcto funcionamiento del juego de instrucciones del C-64.
(108 páginas, tamaño 13,5 x 21,5).

CUPON DE PEDIDO

enviar a:

infodis, s.a.

C/BRAVO MURILLO, 377
28020 MADRID

COPIE O RECORTE ESTE BOLETIN DE PEDIDO.

DESEO RECIBIR LOS SIGUIENTES TITULOS:
15 HORAS CON EL SPECTRUM (P.V.P. 750) ☐
LOS MEJORES PROGRAMAS PARA EL ZX SPECTRUM (P.V.P. 900) ☐
LOS MEJORES PROGRAMAS PARA EL COMMODORE 64 (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL I (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL II (P.V.P. 800) ☐
(más 100 ptas. de gastos de envío).

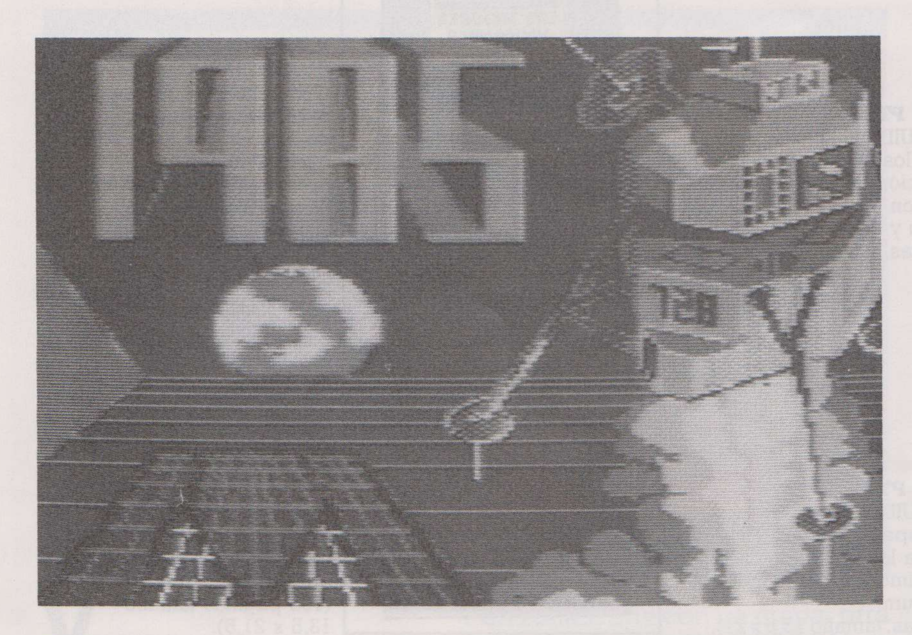
El importe lo abonaré POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO ☐ American Express ☐ Visa ☐ Interbank ☐

Número de mi tarjeta: ________________

NOMBRE ________________
CALLE ________________
CIUDAD ________________
PROVINCIA ________________ C. P. ________________



1985, The Day After



Un título sugerente, ¿verdad? Nos recuerda a la película «El día después», que pudimos ver en las carteleras de cine hace un par de años.

Este es un juego, cuyo argumento está basado en ese hipotético día después del holocausto. Donde tú, en tu nave, eres el encargado de buscar esa poca energía aprovechable para los supervivientes.

Tendrás que ir a cuatro planetas vecinos, en busca de esa preciada energía, pero para ello antes deberás salir de la base en que te encuentras. Poco a poco conseguirás acostumbrarte a los complicados mandos de la nave que pilotas, que por la altísima sensibilidad que tienen a los movimientos de tus manos, te costará un poco controlar.

Antes de adentrarte en el espacio hacia esos planetas vecinos, procura recoger las esferas de energía colocadas en las distintas estaciones espaciales que vayas encontrando.

El principal peligro con que te enfrentas es tu falta de habilidad para manejar el aparato que pilotas. Una vez que lo domines, conseguirás obtener toda la energía necesaria para tus amigos supervivientes del desastre.

Los planetas son de diferentes colores, y en cada uno las dificultadades son distintas. Laberintos más o menos intrincados, o lugares más o menos accesibles en los que obtener lo que buscas.

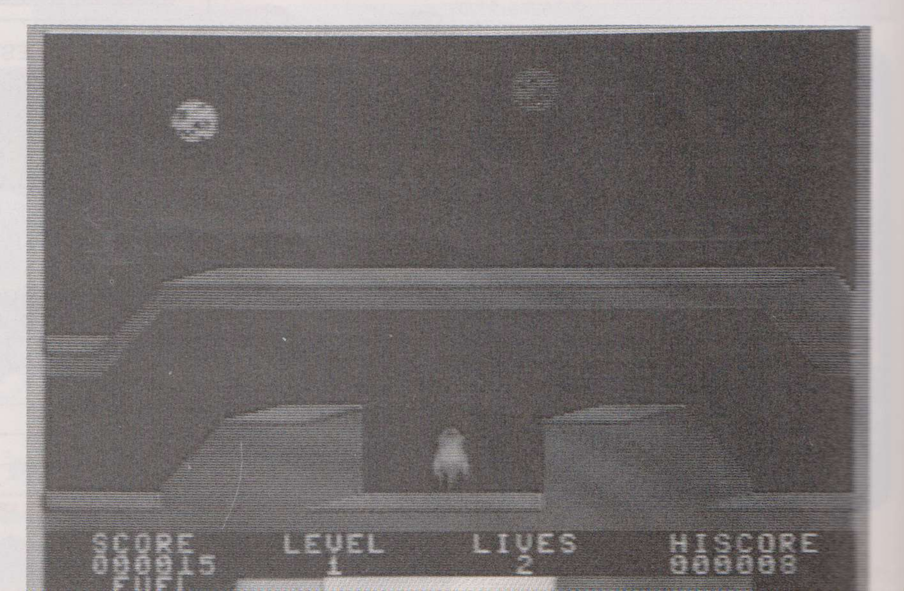
Mediante un potente rayo, del que está dotada tu nave, podrás hacerte con esas esferas que contienen la energía.

En tu planeta no dudan de tu capacidad para resolver la misión, así que ten un poco de paciencia y no los defraudarás.

FICHA DEL JUEGO

NOMBRE:	**THE DAY AFTER**
DE:	**MASTERTRONIC (ABC Soft)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK O TECLADO**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD				■						
ADICCION					■					
GRAFICOS							■			
SONIDO					■					
GENERAL					■					



SU PROGRAMA PARA CUALQUIER SISTEMA COMMODORE PUEDE HACERLE GANAR 5.000 PTAS.

EL PRESENTE CONCURSO ESTA ABIERTO A TODOS NUESTROS LECTORES Y SU PARTICIPACION E INSCRIPCION ES GRATUITA. LEA LAS BASES DEL CONCURSO

- NO SE ESTABLECEN LIMITACIONES EN CUANTO A EXTENSION, TEMA ELEGIDO O MODELO DE ORDENADOR
- LOS CONCURSANTES DEBERAN ENVIARNOS A LA DIRECCION QUE FIGURA AL PIE, EL CASSETTE O DISKETTE CONTENIENDO EL PROGRAMA, UNA EXPLICACION DEL MISMO Y, AL SER POSIBLE, UN LISTADO EN PAPEL DE IMPRESORA. SE PODRAN ENVIAR TANTOS PROGRAMAS COMO SE DESEE
- LOS PROGRAMAS, PREVIA SELECCION, SERAN PUBLICADOS EN LA REVISTA, OBTENIENDO TODOS ELLOS 5.000 PTAS.
- LA DECISION SOBRE LA PUBLICACION O NO DE UN PROGRAMA CORRESPONDE UNICAMENTE AL JURADO NOMBRADO AL EFECTO POR "COMMODORE MAGAZINE", SIENDO SU FALLO INAPELABLE
- LOS CRITERIOS DE SELECCION SE BASARAN EN LA CREATIVIDAD DEL TEMA ELEGIDO Y LA ORIGINALIDAD Y/O SENCILLEZ EN EL METODO DE PROGRAMACION GLOBAL
- ENVIAR A: CONCURSO COMMODORE MAGAZINE



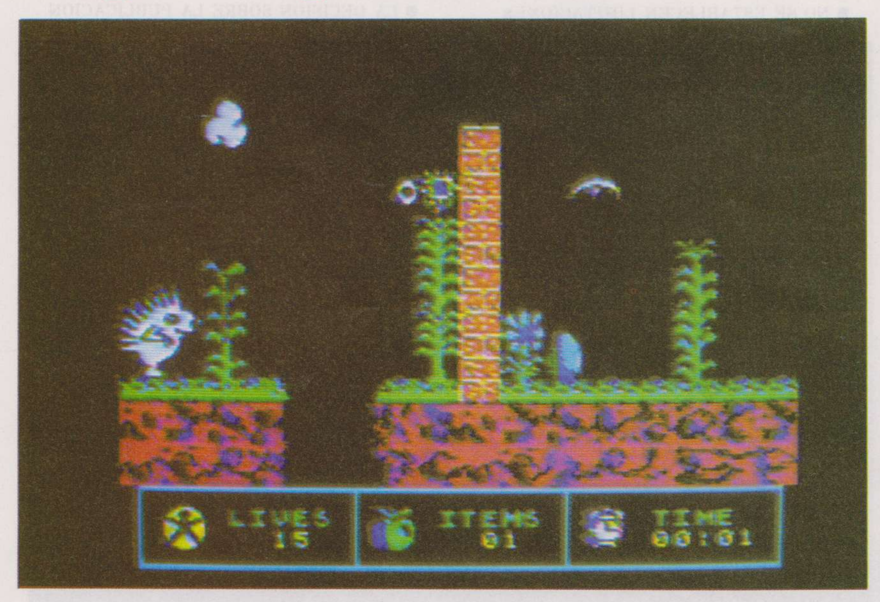
commodore Magazine
C/BRAVO MURILLO, 377 5.° A 28020 MADRID


MAPA


HABILIDAD


AVENTURAS

Spiky Harold



FICHA DEL JUEGO
NOMBRE: **SPIKY HAROLD**
DE: **FIREBIRD (Drosoft)**
ORDENADOR: **COMMODORE-64, 128**
CONTROL: **JOYSTICK, TECLADO**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD									■	
ADICCION								■		
GRAFICOS								■		
SONIDO								■		
GENERAL								■		

Esta vez FIREBIRD nos encomienda la tarea de ayudar a un pequeño erizo SPIKY HAROLD, a que reúna toda la comida que pueda para que cuando llegue el invierno, no le pille sin provisiones en su despensa.

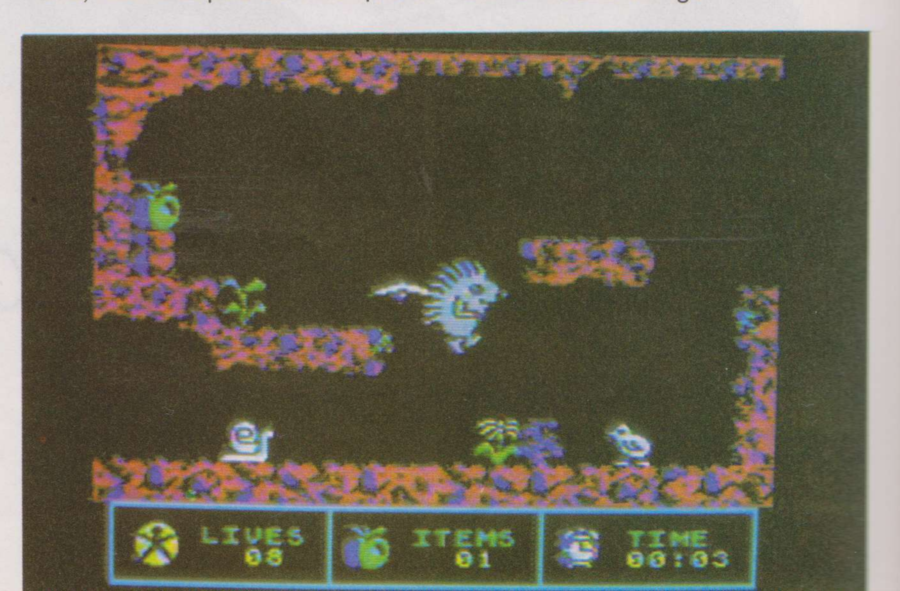
Deberá buscar por todos los sitios posibles. Son 57 cámaras las que tiene que recorrer, pero también son muchos los problemas que tiene que solucionar, el principal es que sólo le quedan 24 horas para que llegue el invierno y ha de darse mucha prisa.

A lo largo de todo su camino, irá encontrando la comida pero también irá encontrando enemigos que no le dejarán que se la lleve, habrá copas de vino que le emborracharán si se descuida, nubes de azufre con las que no se puede chocar cuando salte, goterás que le quitarán una vida, aunque no todo es tan malo para el pobre SPIKY, en su recorrido puede encontrar monedas que le darán más vidas.

En la parte inferior de la pantalla aparece el número de vidas que el quedan, el número de piezas de comida que va guardando en su despensa y un reloj que le indica lo que le falta para que llegue el invierno. Los gráficos están bien logrados, presentan una cuevas llenas de pequeños personajes, la mayoría poco amigos de SPIKY. El sonido se puede quitar o poner, según prefieras, aunque siempre quedarán las pisadas de nuestro protagonista.

Es un juego bastante entretenido, ameno al intentar cada vez descubrir algo distinto que la vez anterior, requiere práctica hasta que consigues salvar todos los obstáculos que te ponen delante, pero al fin lo conseguirás. Puedes usar el *joystick* o bien el teclado definiendo tú, en el menú principal, las teclas que quieres usar, o si no utilizar las que ya vienen definidas en el programa.

De cualquier modo te esperan muchas sorpresas en compañía de tu nuevo amigo.





Park Patrol



Nos encontramos en PAPATOETOE PARK, un parque un tanto especial, ya que estrena guardabosques y aún no está muy familiarizado con sus quehaceres, tú deberás ayudarle para que todo marche perfectamente.

No te creas que tu misión es fácil, hay infinidad de problemas que deberás resolver, hormigas que te robarán la comida cuando estés distraído, troncos flotando y nadadores que chocarán con tu balsa, serpientes que debes evitar, tortugas que te hacen tropezar y te impiden recoger toda la basura que hay esparcida por el parque.

Cuando creas que ya tienes el parque organizado y por si te parece poco, los nadadores en el momento más inoportuno se ahogarán y tendrás que dejar todo para ir a rescatarlos.

En la parte inferior de la pantalla aparece la cantidad de basura que tendrás que recoger para pasar de nivel, por cada pieza de basura que recojas ganarás puntos, así como cada vez que consigas bailar encima de los troncos, también te puedes librar de las hormigas dándoles una patada por detrás.

Hay cuatro niveles que tendrás que superar para ser un buen guardabosques, en el menú principal podrás elegir las dificultades de cada nivel, añadiendo o quitando obstáculos.

También tienes la opción de elegir uno o dos jugadores, y si quieres que sean hombres o mujeres, para ello usarás el *joystick*. Los gráficos de este juego están bien conseguidos, mostrándonos el parque con su pantano y su caseta del guarda, el sonido es un poco más normal. En conjunto es un juego que entretiene y no se hace monótono, es original en cuanto al tema y puede hacerte pasar un buen rato frente a tu ordenador.

Conforme vayas jugando irás descubriendo cosas nuevas, sólo tendrás que fijarte en todo lo que te rodea y evitar los tropezones.

FICHA DEL JUEGO

NOMBRE: **PARK PATROL**
DE: **FIREBIRD (Drosoft)**
ORDENADOR: **COMMODORE-64, 128**
CONTROL: **JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD								■		
ADICCION							■			
GRAFICOS								■		
SONIDO						■				
GENERAL							■			



¿te interesa?

Vendo ordenador C-64 y unidad de disco 1541 con poco uso en 65.000 ptas. Regalo *joystick* Quickshot II y cinco discos con juegos. Dirigirse a: Bruno Celaya Montero. C/ Larraina, 19, 5ºD. 31011 Pamplona. Tel.: (948) 26 49 81.

●

Vendo juegos originales C-64 a 500 y 1.000 ptas. De la calidad de Green Beret o Kane. Interesados facilito lista de los mismos. Dirigirse a: Bruno Celaya Montero. C/ Larraina, 19, 5ºD. 31011 Pamplona. Tel.: (948) 26 49 81.

●

Necesito recaudar fondos para la compra de una impresora, por lo que vendo: monitor fósforo verde (no molesta vista) por 17.000. Un *joystick* Auto Fire por 1.700 ptas. Varias revistas de Commodore por 250 c/u. Y varios juegos, entre ellos Hardball, Tour de Francia, Rambo, Beach Head II... Por 1.500 c/u. Interesados dirigirse a: David Carmona Salas. C/ Gabriel Matute, 8, 5ºF. 11008 Cádiz. Tel.: (956) 25 72 19.

●

Cambio por C-64 + unidad de discos 1541 un equipo CB S.Star B600, fuente, antena, acoplador, 5WR, cables, embalaje orig., facturas. Valor 45.000. Pago la diferencia. Sólo Barcelona y provincias. Contactar con Gabriel en el tel.: (93) 217 90 80.

●

Por cambio de equipo vendo impresora MPS 801, prácticamente nueva, por 25.000 ptas. Interesados dirigirse a: Galo Sánchez Casado. Cardoner, 42-44, 2. 08024 Barcelona. Tel.: (93) 210 31 11.

●

Vendo un lote de cintas para el CBM-64 entre las que se encuentran: Kung Fu Master, Rambo, Pitstop II, Decathlon, Space Shutle, Skyfox, Mach Point, Bruce-Lee, Commando y 180 programas igual de buenos, con utilidades y funcionando. También vendo *joystick* Quick Shot II nuevo, un cartucho valorado en 6.000 ptas. «HES MON» y 50 revistas de informática. Lo vendo todo o lo cambio por una unidad de discos para CBM-64 o una impresora en buen estado de uso. También doy 6 libros de informática. Interesados llamar durante todo el día al tel.: (93) 318 44 61 o escribir a: Vicente Gascón. C/ Lancaster, 16, 1º. 08001 Barcelona.

●

¡¡Atención!! Vendo juegos para C-64. No importa que tengas pocos. Mándame tu lista y prometo mandarte la mía. Mi dirección es: Oscar Fernández Orallo. Plza. la Fortaleza II, 4ºC. 24400 Ponferrada. León.

Si eres de Ponferrada y quieres formar un club, llámame. Tel.: (987) 41 85 73.

●

Se ha formado en Barcelona el Club B.C.S. (Barna Computer Service). Abarca Spectrum, Commodore, Amstrad y MSX. Interesados escribir al apartado de correos de Barcelona nº 2309. C.P. 08080 (indicando modelo de ordenador) o llamar al tel.: (93) 309 56 52. Preguntar por Angel.

●

Intercambio programas en disco para CBM-64. Poseo más de 1.000. Prometo contestar. Mandar lista a Antonio Molina García. Barriada de la Constitución, bloque 12, 2ºC. Melilla.

●

Para recaudar fondos para la compra de un nuevo equipo (C-128) vendo para el C-64 lo siguiente: entre juegos y utilidades hacen un total de 250 programas, juegos tales como Commando, One on One, Fist I, Breakdance, The Dambusters, etc., utilidades tales como Tasword, Simon's Basic, Sam Reciter, etc. También vendo revistas de Commodore World, Magazine, Input, Tu micho, Muy ordenadores, Ordenador Personal, etc. Y por último vendo libros: Gráficos y sonido para el C-64, varios libros de informática, Diccionario de electrónica e informática, etc. Interesados dirigirse a: Miguel Angel Huelves García. C/ Entre Arroyos, 64, 1ºB. 28030 Madrid. Tel.: (91) 439 96 31.

●

Vendo ordenador Vic-20 y unidad de *cassette* nuevos. Regalo revistas y cintas. Interesados dirigirse a: Félix Tajuelo. Islas Cíes, 13. 28035 Madrid. Tel.: (91) 739 33 68. Precio a convenir.

Cambio módulo expansor para 6 cartuchos Vic-1020, dos cartuchos (8 y 16 K) para Vic 20 por cualquier material informático, fotográfico o electrónico, etc. Dirigirse a: José Casais Blanco. Ponte de Saa, 74. Villalba 27800 Lugo.

●

Tengo un CBM-64 y un *cassette* C2N. Intercambio todo tipo de programas. Aunque mi lista no es muy extensa, todos mis programas son de calidad y últimas novedades (Infiltrator, Mission Elevator, Spindizzy, etc.). Mi dirección es: David Rincón Rivera. Ctra. D'Esplugues, 66, 2º-1ª. Cornellá de Llobregat. 08940 Barcelona. Prometo contestar.

●

Vendo ordenador C-128 por 45.000 ptas. Monitor fósforo verde con cable 40/80 columnas por 15.000 ptas. *Datassette,* 5.000 ptas. Unidad de disco 1541, 30.000 ptas. Unidad 1571, 45.000 ptas. Contactar con Luis E. Gutiérrez Behemerid. C/ Labradores, 30, 6ºB. 47004 Valladolid. Tel.: (983) 20 23 64.

●

Cambio 50 juegos para el Commodore 64 (la mayoría con Turbo) de entre los cuales destacan: Zorro, Skool Daze, Dinamite Dan, Blue Max, Dambusters, Staff of Karnath, High Noon, Falcon Patrol II, y otros muchos. Interesados dirigirse a: Vicente Ramos. C/ S. Ramón, 11, Burriana (Castellón).

●

Club Usuaris Commodore-Tarragona, centro homologado por la Generalitat de Catalunya; nuestras actividades están distribuidas en distintas secciones: Biblioteca, Programoteca, Cursos, Asesoramiento, etc. Dos años al servicio de nuestros socios. Para información: Club Usuaris Commodore. Apartado de Correos 176. Tarragona. O también, c/ Fortuny, 4, 2.º 2.ª Tarragona.

●

Usuario de CBM-64, desea contactar con otros usuarios interesados en las comunicaciones por Ordenador, a través de *Modem* telefónico. Interesados llamar al

(977) 86 00 63, o escribir a Eduard Macip. C/ Forn de la vila, 5. 43400 Montblanc (Tarragona). También cambio programas.

Vendo junto o por separado videojuego Atari, dos *paddles,* dos *joysticks,* dos teclados numéricos con entrada Atari, 15 cartuchos de juegos. Precio a convenir. Dirigirse a: Manuel Castaño Cano. Avda. Teodomiro, 1, 1.° C. 03300 Orihuela (Alicante).

Me interesaría vender la colección completa de revistas Commodore Magazine (del 1 al 35), y también el Curso de introducción al Basic I y II, de Andrew Colin (o cambiar).

También vendo la cinta original del juego de ajedrez, Gran Master 64, y el cartucho International Soccer. Para ofertas dirigirse a: Jordi Bonell Busquets. Plaza del Progres, 3. Planta baja. Rubi 08191. Vallés Occidental (Barcelona).

Me gustaría adquirir o cambiar programas de dibujo y diseño técnico. También programas de cálculos mecánicos o de ingeniería para Commodore 64. Interesados dirigirse a: Jordi Bonell Busquets. Plaza del Progres, 3. Planta baja. Rubi 08191. Vallés Occidental (Barcelona).

Urge comprar unidad de discos. O puedo ofrecer un lote de programas valorado en 130.000 ptas. a cambio de unidad de discos o de impresora o de monitor color o de cualquier periférico para CBM-64. Cambio CBM-64, por CBM-128 pagando la diferencia y regalo programas. Vendo videopac Phiplips G-7400 convertible en ordenador, regalo programas. Intercambio *software* para CBM, tengo todas o casi todas las últimas novedades. Dirigirse a: Francisco Javier Bernal Malillos. C/ Echegaray, 1. Cantalejo (Segovia). Tel.: (911) 52 04 04.

Vendo compilador de COBOL, Lenguaje «C» y SPEED Termina Program para Commodore 128, en modo 128.

Compiladores COBOL, Lenguaje «C», Lenguaje «ADA» y PASCAL para Commodore 64. No hace falta CP/M.

Gastos de envío por mi cuenta. Fernando Gallardo R. Apdo. 315. 06800 Mérida (Badajoz). Tel.: (924) 31 07 61.

Vendo VIC-20 con *datassette* y ampliación memoria; junto con curso de programación y juegos. Todo por 15.000 ptas. Dirigirse a José María. Tel.: (91) 202 90 33.

ANUNCIOS GRATUITOS

Todos los anuncios (compras, ventas, cambios o comunicaciones de clubs de usuarios, etc.) que van en esta sección, deben tener un máximo de cuarenta palabras. Con el fin de facilitar la transcripción de los anuncios hemos recuadrado cuarenta espacios para que en cada uno vaya una palabra. Después, recortar y mandar a:

commodore Magazine
ANUNCIOS GRATUITOS
C/ Bravo Murillo, 377, 5.° A
28020 MADRID

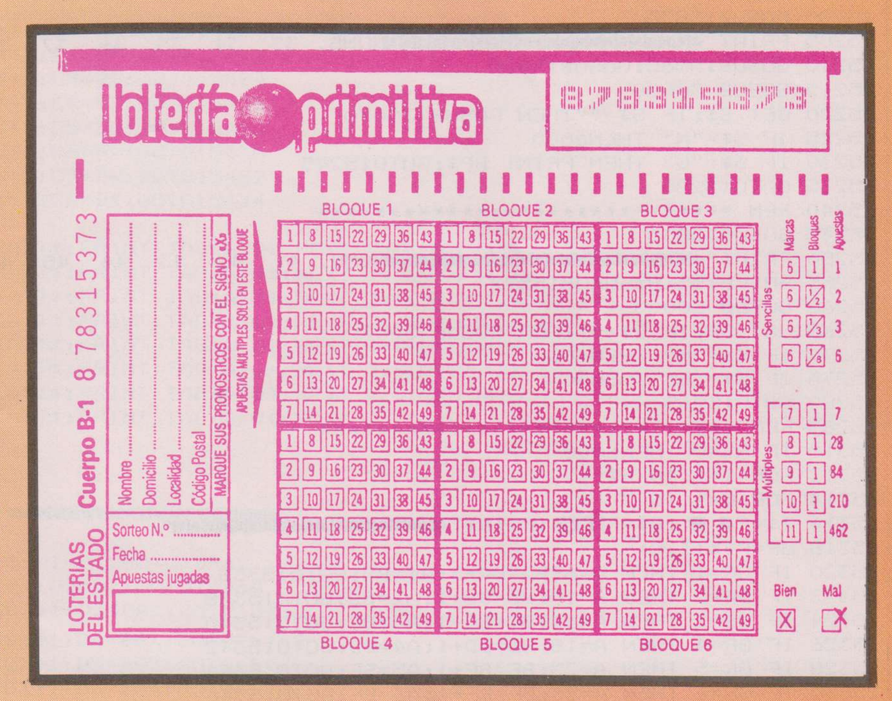
Estadística Loto

2ª PARTE

Esta es la segunda parte del programa enviado por Jesús García Lázaro para el CBM-64 que te permite disponer semanalmente de una Estadística de la Lotería Primitiva.

Las instrucciones de uso están junto a la primera parte que puedes encontrar en el número anterior.

La longitud del programa nos obligó a dividirlo en dos partes, y aunque por su extensión no debiera haber superado las pruebas de selección, el interés que despierta la Lotería Primitiva y la completa y fácil estructuración del programa, nos indujo a su publicación.




```
15035 GOSUB 30000
15040 GET S$:IF S$=""THEN GOSUB 30070
15045 IF S$="M" THEN6000
15050 IF S$="G" THEN PRINT BP$:GOTO15065
15055 GOTO15040
15060 REM ******************************
15065 GOSUB15835
15070 PRINT"SQQQQQQQQQQQQQQQQQQQQNUMS. 9   10  11  12  13  14  15  16"
15075 GOSUB15568:GOSUB15354
15080 GOSUB 30000
15085 GET S$:IF S$=""THEN GOSUB 30070
15090 IF S$="M" THEN6000
15095 IF S$="G" THEN PRINT BP$:GOTO15110
15100 GOTO15085
15105 REM ******************************
15110 GOSUB15835
15115 PRINT"SQQQQQQQQQQQQQQQQQQQQNUMS. 17  18  19  20  21  22  23  24"
15120 GOSUB15568:GOSUB15396
15125 GOSUB 30000
15130 GET S$:IF S$=""THEN GOSUB 30070
15135 IF S$="M" THEN6000
15140 IF S$="G" THEN PRINT BP$:GOTO15155
15145 GOTO15130
15150 REM ******************************
15155 GOSUB15835
```

PROGRAMAS

```
15160 PRINT"SQQQQQQQQQQQQQQQQQQQQQNUMS. 25   26   27   28   29   30   31   32"
15165 GOSUB15568:GOSUB15438
15170 GOSUB 30000
15175 GET S$:IF S$=""THEN GOSUB 30070
15180 IF S$="M" THEN6000
15185 IF S$="G" THEN PRINT BP$:GOTO15200
15190 GOTO15175
15195 REM ****************************
15200 GOSUB15835
15205 PRINT"SQQQQQQQQQQQQQQQQQQQQQNUMS. 33   34   35   36   37   38   39   40"
15210 GOSUB15568:GOSUB15480
15215 GOSUB 30000
15220 GET S$:IF S$=""THEN GOSUB 30070
15225 IF S$="M" THEN6000
15230 IF S$="G" THEN PRINT BP$:GOTO15245
15235 GOTO15220
15240 REM ****************************
15245 GOSUB15835
15250 PRINT"SQQQQQQQQQQQQQQQQQQQQQNUMS. 41   42   43   44   45   46   47   48   49"
15255 GOSUB15568:GOSUB15522
15260 GOSUB 30000
15265 GET S$:IF S$=""THEN GOSUB 30070
15270 IF S$="M" THEN6000
15275 IF S$="G" THEN6000
15280 GOTO15265
15285 REM ****************************
15310 IF R%=0 THEN6000
15312 FOR I=1 TO R%
15314 FOR C=1 TO 7
15316 T$="▜ ▀":Q$="TTT":PRINT"SQQQQQQQQQQQQQQQQ"
15318 GR=F1%(I,C)
15320 IF GR=1 THEN A=6:BA=BA+1:A1=BA:GOTO15336
15322 IF GR=2 THEN A=10:BB=BB+1:A2=BB:GOTO15338
15324 IF GR=3 THEN A=14:BC=BC+1:A3=BC:GOTO15340
15326 IF GR=4 THEN A=18:BD=BD+1:A4=BD:GOTO15342
15328 IF GR=5 THEN A=22:BE=BE+1:A5=BE:GOTO15344
15330 IF GR=6 THEN A=26:BF=BF+1:A6=BF:GOTO15346
15332 IF GR=7 THEN A=30:BG=BG+1:A7=BG:GOTO15348
15334 IF GR=8 THEN A=34:BH=BH+1:A8=BH:GOTO15350
15335 GOTO 15352
15336 FOR Z=1 TO BA:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15352
15338 FOR Z=1 TO BB:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15352
15340 FOR Z=1 TO BC:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15352
15342 FOR Z=1 TO BD:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15352
15344 FOR Z=1 TO BE:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15352
15346 FOR Z=1 TO BF:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15352
15348 FOR Z=1 TO BG:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15352
15350 FOR Z=1 TO BH:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z
15352 NEXT C,I:RETURN
15354 FOR I=1 TO R%
15356 FOR C=1 TO 7
15358 T$="▜ ▀":Q$="TTT":PRINT"SQQQQQQQQQQQQQQQQ"
15360 GR=F1%(I,C)
15362 IF GR=9 THEN A=6:BI=BI+1:B1=BI:GOTO15378
15364 IF GR=10 THEN A=10:BJ=BJ+1:B2=BJ:GOTO15380
15366 IF GR=11 THEN A=14:BK=BK+1:B3=BK:GOTO15382
15368 IF GR=12 THEN A=18:BL=BL+1:B4=BL:GOTO15384
15370 IF GR=13 THEN A=22:BM=BM+1:B5=BM:GOTO15386
15372 IF GR=14 THEN A=26:BN=BN+1:B6=BN:GOTO15388
15374 IF GR=15 THEN A=30:BO=BO+1:B7=BO:GOTO15390
15376 IF GR=16 THEN A=34:BP=BP+1:B8=BP:GOTO15392
15377 GOTO 15394
15378 FOR Z=1 TO BI:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15394
15380 FOR Z=1 TO BJ:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15394
15382 FOR Z=1 TO BK:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15394
15384 FOR Z=1 TO BL:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15394
```

```
15386 FOR Z=1 TO BM:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15394
15388 FOR Z=1 TO BN:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15394
15390 FOR Z=1 TO BO:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15394
15392 FOR Z=1 TO BP:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z
15394 NEXT C,I:RETURN
15396 FOR I=1 TO R%
15398 FOR C=1 TO 7
15400 T$="▚ ▀":Q$="TTT":PRINT"SQQQQQQQQQQQQQQQQQ"
15402 GR=F1%(I,C)
15404 IF GR=17 THEN A=6:BQ=BQ+1:C1=BQ:GOTO15420
15406 IF GR=18 THEN A=10:BR=BR+1:C2=BR:GOTO15422
15408 IF GR=19 THEN A=14:BS=BS+1:C3=BS:GOTO15424
15410 IF GR=20 THEN A=18:BT=BT+1:C4=BT:GOTO15426
15412 IF GR=21 THEN A=22:BU=BU+1:C5=BU:GOTO15428
15414 IF GR=22 THEN A=26:BV=BV+1:C6=BV:GOTO15430
15416 IF GR=23 THEN A=30:BW=BW+1:C7=BW:GOTO15432
15418 IF GR=24 THEN A=34:BX=BX+1:C8=BX:GOTO15434
15419 GOTO 15436
15420 FOR Z=1 TO BQ:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15436
15422 FOR Z=1 TO BR:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15436
15424 FOR Z=1 TO BS:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15436
15426 FOR Z=1 TO BT:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15436
15428 FOR Z=1 TO BU:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15436
15430 FOR Z=1 TO BV:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15436
15432 FOR Z=1 TO BW:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15436
15434 FOR Z=1 TO BX:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z
15436 NEXT C,I:RETURN
15438 FOR I=1 TO R%
15440 FOR C=1 TO 7
15442 T$="▚ ▀":Q$="TTT":PRINT"SQQQQQQQQQQQQQQQQQ"
15444 GR=F1%(I,C)
15446 IF GR=25 THEN A=6:BY=BY+1:D1=BY:GOTO15462
15448 IF GR=26 THEN A=10:BZ=BZ+1:D2=BZ:GOTO15464
15450 IF GR=27 THEN A=14:BA%=BA%+1:D3=BA%:GOTO15466
15452 IF GR=28 THEN A=18:BB%=BB%+1:D4=BB%:GOTO15468
15454 IF GR=29 THEN A=22:BC%=BC%+1:D5=BC%:GOTO15470
15456 IF GR=30 THEN A=26:BD%=BD%+1:D6=BD%:GOTO15472
15458 IF GR=31 THEN A=30:BE%=BE%+1:D7=BE%:GOTO15474
15460 IF GR=32 THEN A=34:BF%=BF%+1:D8=BF%:GOTO15476
15461 GOTO 15478
15462 FOR Z=1 TO BY:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15478
15464 FOR Z=1 TO BZ:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15478
15466 FOR Z=1 TO BA%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15478
15468 FOR Z=1 TO BB%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15478
15470 FOR Z=1 TO BC%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15478
15472 FOR Z=1 TO BD%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15478
15474 FOR Z=1 TO BE%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15478
15476 FOR Z=1 TO BF%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z
15478 NEXT C,I:RETURN
15480 FOR I=1 TO R%
15482 FOR C=1 TO 7
15484 T$="▚ ▀":Q$="TTT":PRINT"SQQQQQQQQQQQQQQQQQ"
15486 GR=F1%(I,C)
15488 IF GR=33 THEN A=6:BG%=BG%+1:E1=BG%:GOTO15504
15490 IF GR=34 THEN A=10:BH%=BH%+1:E2=BH%:GOTO15506
15492 IF GR=35 THEN A=14:BI%=BI%+1:E3=BI%:GOTO15508
15494 IF GR=36 THEN A=18:BJ%=BJ%+1:E4=BJ%:GOTO15510
15496 IF GR=37 THEN A=22:BK%=BK%+1:E5=BK%:GOTO15512
15498 IF GR=38 THEN A=26:BL%=BL%+1:E6=BL%:GOTO15514
15500 IF GR=39 THEN A=30:BM%=BM%+1:E7=BM%:GOTO15516
15502 IF GR=40 THEN A=34:BN%=BN%+1:E8=BN%:GOTO15518
15503 GOTO 15520
15504 FOR Z=1 TO BG%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15520
15506 FOR Z=1 TO BH%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15520
15508 FOR Z=1 TO BI%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15520
15510 FOR Z=1 TO BJ%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15520
```

PROGRAMAS

```
15512 FOR Z=1 TO BK%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15520
15514 FOR Z=1 TO BL%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15520
15516 FOR Z=1 TO BM%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15520
15518 FOR Z=1 TO BN%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z
15520 NEXT C,I:RETURN
15522 FOR I=1 TO R%
15524 FOR C=1 TO 7
15526 T$="N ■":Q$="TTT":PRINT"SQQQQQQQQQQQQQQQQQ"
15528 GR=F1%(I,C)
15530 IF GR=41 THEN A=6:BO%=BO%+1:F1=BO%:GOTO15548
15532 IF GR=42 THEN A=10:BP%=BP%+1:F2=BP%:GOTO15550
15534 IF GR=43 THEN A=14:BQ%=BQ%+1:F3=BQ%:GOTO15552
15536 IF GR=44 THEN A=18:BR%=BR%+1:F4=BR%:GOTO15554
15538 IF GR=45 THEN A=22:BS%=BS%+1:F5=BS%:GOTO15556
15540 IF GR=46 THEN A=26:BT%=BT%+1:F6=BT%:GOTO15558
15542 IF GR=47 THEN A=30:BU%=BU%+1:F7=BU%:GOTO15560
15544 IF GR=48 THEN A=34:BV%=BV%+1:F8=BV%:GOTO15562
15546 IF GR=49 THEN A=38:BW%=BW%+1:F9=BW%:GOTO15564
15547 GOTO 15566
15548 FOR Z=1 TO BO%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15566
15550 FOR Z=1 TO BP%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15566
15552 FOR Z=1 TO BQ%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15566
15554 FOR Z=1 TO BR%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15566
15556 FOR Z=1 TO BS%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15566
15558 FOR Z=1 TO BT%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15566
15560 FOR Z=1 TO BU%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15566
15562 FOR Z=1 TO BV%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z:GOTO15566
15564 FOR Z=1 TO BW%:PRINT TAB(A)T$:PRINT TAB(A)Q$:NEXT Z
15566 NEXT C,I:RETURN
15568 REM ******************************
15570 BA=0:BB=0:BC=0:BD=0:BE=0:BF=0:BG=0:BH=0:BI=0:BJ=0:BK=0:BL=0:BM=0:BN=0
15572 BO=0:BP=0:BQ=0:BR=0:BS=0:BT=0:BU=0:BV=0:BW=0:BX=0:BY=0:BZ=0
15574 BA%=0:BB%=0:BC%=0:BD%=0:BE%=0:BF%=0:BG%=0:BH%=0:BI%=0:BJ%=0:BK%=0:BL%=0
15576 BM%=0:BN%=0:BO%=0:BP%=0:BQ%=0:BR%=0:BS%=0:BT%=0:BU%=0:BV%=0:BW%=0
15578 RETURN
15580 REM *****************************
15590 REM ***  DIBUJO DEL GRAFICO   ***
15830 REM *****************************
15832 :
15835 PRINT"TN.VECES"
15840 PRINT"   19|────────────────────────────────────"
15845 PRINT"   18|────────────────────────────────────"
15850 PRINT"   17|────────────────────────────────────"
15855 PRINT"   16|────────────────────────────────────"
15860 PRINT"   15|────────────────────────────────────"
15865 PRINT"   14|────────────────────────────────────"
15870 PRINT"   13|────────────────────────────────────"
15875 PRINT"   12|────────────────────────────────────"
15880 PRINT"   11|────────────────────────────────────"
15885 PRINT"   10|────────────────────────────────────"
15890 PRINT"    9|────────────────────────────────────"
15895 PRINT"    8|────────────────────────────────────"
15900 PRINT"    7|────────────────────────────────────"
15905 PRINT"    6|────────────────────────────────────"
15910 PRINT"    5|────────────────────────────────────"
15915 PRINT"    4|────────────────────────────────────"
15920 PRINT"    3|────────────────────────────────────"
15925 PRINT"    2|────────────────────────────────────"
15930 PRINT"    1|────────────────────────────────────"
15935 PRINT"      ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━"
15940 PRINT"SQQQQQQQQQQQQQQQQQQQQQQQQ   ESPERE POR FAVOR, PROCESANDO DATOS"
15945 RETURN
16000 REM *****************************
16010 REM SALIDA DE DATOS POR IMPRESORA
16020 REM *****************************
16030 :
```

```
16040 PRINT BP$:POKE 53280,0:POKE 53281,0:POKE 19,1
16042 INPUT"■■■   FECHA EN DD/MM/AA > ";F$
16044 PRINT BP$
16050 OPEN4,4,0:CLOSE4
16060 IF ST<>0 THEN 16080
16070 GOTO 16120
16080 PRINT"■■■■■ CONECTE LA IMPRESORA Y PULSE UNA TECLA"
16090 VV=212*256:POKE VV+24,15:POKE VV,0:POKE VV+1,80:POKE VV+5,9:POKE VV+4,0
16100 POKE VV+4,33:FOR M=1 TO 200:NEXTM
16110 GET S$:IF S$=""THEN 16090
16120 OPEN4,4:OPEN 3,3
16130 FOR Z=1 TO 40:PRINT#3,"‾";:NEXT
16140 PRINT#3,"■■■  IMPRIMIENDO DATOS * ESPERE POR FAVOR"
16150 FOR Z=1 TO 40:PRINT#3,"‾";:NEXT
16160 GOSUB 19000:GOSUB 18000:PRINT#4,CHR$(27)"E"
16164 PRINT#4,CHR$(27)"G"
16166 PRINT#4,"    * NUMEROS IMPARES *"
16168 PRINT#4,CHR$(27)"H"
16170 PRINT#4,"    TOTAL DE NUMEROS IMPARES ................:"NITAB(46)PA" %"
16174 PRINT#4,CHR$(27)"G"
16176 PRINT#4,"    * NUMEROS PARES *"
16178 PRINT#4,CHR$(27)"H"
16190 PRINT#4,"    TOTAL DE NUMEROS PARES ..................:"NPTAB(46)PB" %"
16194 PRINT#4,CHR$(27)"G"
16196 PRINT#4,"    * TERMINACIONES *"
16198 PRINT#4,CHR$(27)"H"
16210 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 0 ........:"X0TAB(46)PC" %"
16230 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 1 ........:"X1TAB(46)PD" %"
16250 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 2 ........:"X2TAB(46)PE" %"
16270 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 3 ........:"X3TAB(46)PF" %"
16290 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 4 ........:"X4TAB(46)PG" %"
16310 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 5 ........:"X5TAB(46)PH" %"
16320 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 6 ........:"X6TAB(46)PI" %"
16330 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 7 ........:"X7TAB(46)PJ" %"
16340 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 8 ........:"X8TAB(46)PK" %"
16350 PRINT#4,"    TOTAL DE NUMEROS TERMINADOS EN 9 ........:"X9TAB(46)PL" %"
16352 PRINT#4,CHR$(27)"G"
16354 PRINT#4,"    * NUMEROS POR DECENAS *"
16356 PRINT#4,CHR$(27)"H"
16410 PRINT#4,"    TOTAL DE NUMEROS DE LA PRIMERA DECENA ...:"N1TAB(46)PM" %"
16420 PRINT#4,"    TOTAL DE NUMEROS DE LA SEGUNDA DECENA ...:"N2TAB(46)PN" %"
16440 PRINT#4,"    TOTAL DE NUMEROS DE LA TERCER DECENA ....:"N3TAB(46)PO" %"
16460 PRINT#4,"    TOTAL DE NUMEROS DE LA CUARTA DECENA ....:"N4TAB(46)PP" %"
16480 PRINT#4,"    TOTAL DE NUMEROS DE LA QUINTA DECENA ....:"N5TAB(46)PQ" %"
16494 PRINT#4,CHR$(27)"G"
16496 PRINT#4,"    * FILAS HORIZONTALES *"
16498 PRINT#4,CHR$(27)"H"
16500 PRINT#4,"    TOTAL DE LA PRIMERA FILA HORIZONTAL .....:"L1TAB(46)PR" %"
16520 PRINT#4,"    TOTAL DE LA SEGUNDA FILA HORIZONTAL .....:"L2TAB(46)PS" %"
16540 PRINT#4,"    TOTAL DE LA TERCERA FILA HORIZONTAL .....:"L3TAB(46)PT" %"
16560 PRINT#4,"    TOTAL DE LA CUARTA FILA HORIZONTAL ......:"L4TAB(46)PU" %"
16580 PRINT#4,"    TOTAL DE LA QUINTA FILA HORIZONTAL ......:"L5TAB(46)PV" %"
16600 PRINT#4,"    TOTAL DE LA SEXTA FILA HORIZONTAL .......:"L6TAB(46)PW" %"
16620 PRINT#4,"    TOTAL DE LA SEPTIMA FILA HORIZONTAL .....:"L7TAB(46)PX" %"
16624 PRINT#4,CHR$(27)"G"
16626 PRINT#4,"    * FILAS VERTICALES *"
16628 PRINT#4,CHR$(27)"H"
16640 PRINT#4,"    TOTAL DE LA PRIMERA COLUMNA VERTICAL ....:"V1TAB(46)QA" %"
16660 PRINT#4,"    TOTAL DE LA SEGUNDA COLUMNA VERTICAL ....:"V2TAB(46)QB" %"
16680 PRINT#4,"    TOTAL DE LA TERCERA COLUMNA VERTICAL ....:"V3TAB(46)QC" %"
16700 PRINT#4,"    TOTAL DE LA CUARTA COLUMNA VERTICAL .....:"V4TAB(46)QD" %"
16720 PRINT#4,"    TOTAL DE LA QUINTA COLUMNA VERTICAL .....:"V5TAB(46)QE" %"
16740 PRINT#4,"    TOTAL DE LA SEXTA COLUMNA VERTICAL ......:"V6TAB(46)QF" %"
16760 PRINT#4,"    TOTAL DE LA SEPTIMA COLUMNA VERTICAL ....:"V7TAB(46)QG" %"
16764 PRINT#4,CHR$(27)"G"
```

PROGRAMAS

```
16766 PRINT#4,"    * MITAD DE BLOQUES *"
16768 PRINT#4,CHR$(27)"H"
16780 PRINT#4,"     TOTAL NUMEROS DE LA PRIMERA MITAD 1-25 ..:"J1TAB(46)QH" %"
16800 PRINT#4,"     TOTAL NUMEROS DE LA SEGUNDA MITAD 26-49 .:"J2TAB(46)QI" %"
16810 PRINT#4,CHR$(27)"F"
16820 CLOSE 3
16830 CLOSE 4:PRINT BP$:GOTO 6000
18000 REM ****************************
18010 REM ** RUTINA TITULO IMPRESORA **
18020 REM ****************************
18030 :
18040 FOR Z=1 TO 80:PRINT#4,"*";:NEXT Z
18050 PRINT#4,CHR$(27)"E",CHR$(27)"G"
18060 PRINT#4,"                DATOS ESTADISTICOS                        EN FECHA :"F$
18065 PRINT#4,CHR$(27)"H",CHR$(27)"F"
18095 PRINT#4,CHR$(27)"V",CHR$(14)
18100 PRINT#4,"      * * *  LOTERIA PRIMITIVA * * *"CHR$(20),CHR$(27)"V"
18110 FOR Z=1 TO 80:PRINT#4,"=";:NEXT Z
18130 PRINT#4,CHR$(27)"E",CHR$(27)"G"
18140 PRINT#4,"    DENOMINACION                                          DATOS     PORCENTAJE"
18150 PRINT#4,CHR$(27)"H",CHR$(27)"F"
18165 FOR Z=1 TO 80:PRINT#4,"=";:NEXT Z
18170 RETURN
18180 :
19000 REM ****************************
19010 REM *** RUTINA DE PORCENTAJES ***
19020 REM ****************************
19030 K=R%*7:L=100
19040 PA=INT((NI*L)/K):PB=INT((NP*L)/K):PC=INT((X0*L)/K):PD=INT((X1*L)/K)
19045 PE=INT((X2*L)/K)
19050 PF=INT((X3*L)/K):PG=INT((X4*L)/K):PH=INT((X5*L)/K):PI=INT((X6*L)/K)
19055 PJ=INT((X7*L)/K)
19060 PK=INT((X8*L)/K):PL=INT((X9*L)/K):PM=INT((N1*L)/K):PN=INT((N2*L)/K)
19065 PO=INT((N3*L)/K)
19070 PP=INT((N4*L)/K):PQ=INT((N5*L)/K):PR=INT((L1*L)/K):PS=INT((L2*L)/K)
19075 PT=INT((L3*L)/K)
19080 PU=INT((L4*L)/K):PV=INT((L5*L)/K):PW=INT((L6*L)/K):PX=INT((L7*L)/K)
19090 QA=INT((V1*L)/K):QB=INT((V2*L)/K):QC=INT((V3*L)/K):QD=INT((V4*L)/K)
19095 QE=INT((V5*L)/K)
19100 QF=INT((V6*L)/K):QG=INT((V7*L)/K):QH=INT((J1*L)/K):QI=INT((J2*L)/K)
19110 RETURN
25000 REM ****************************
25010 REM ***   RUTINA DE TEXTOS    ***
25020 REM ****************************
25030 :
25050 A$=" PULSA ' M ' PARA MENU DE DATOS  * "
25060 X=LEN(A$)
25070 PRINT"SQQQQQQQQQQQQQQQQQQQQQ"
25080 B$=RIGHT$(A$,X)+LEFT$(A$,1):PRINT"["MID$(B$,1,40)" ] ":A$=B$
25090 FOR Z=0 TO 50:NEXT Z:RETURN
25100 REM ****************************
30000 REM ****************************
30010 REM ***   RUTINA DE TEXTOS    ***
30020 REM ****************************
30030 :
30050 A$=" PULSA ' M ' MENU   ' G ' GRAFICO *"
30060 X=LEN(A$)
30070 PRINT"SQQQQQQQQQQQQQQQQQQQQQQ"
30080 B$=RIGHT$(A$,X)+LEFT$(A$,1):PRINT"["MID$(B$,1,40)"]":A$=B$
30090 FOR Z=0 TO 50:NEXT Z:RETURN
30100 REM ****************************

READY.
```

Looping Ball

Odoric Benavent Mateu, nos envía desde Barcelona un nuevo estilo de aventuras, donde los reflejos hacen el papel más importante en la acción. Empezaréis en el nivel 3, y si tenéis manitas de plata podréis llegar al nivel 10, es decir, al final de la aventura. Breve explicación de algunas de las variables a modificar.

MN$ Contiene el nombre del autor del récord de presentación.

VP Contiene el récord de puntuación.

VI Contiene las vidas del juego. Son 3.

PU Contiene la puntuación que al final será multiplicada por 100.

FO Es el número de *pixel* que el *sprite* recorre en cada pulsación.

PA Informa del nivel donde se está jugando.

TI$ Está preparado para que se esté 20 segundos en cada pantalla.

F Es una de las posiciones del *sprite*.

BA Contiene el valor que hace desplazar la pantalla. Las demás variables no son de gran importancia.

El *joystik* va conectado en el *port* 2.

Estructura

0-22	Rutina CM., *scroll* de pantalla.
23-31	Carga *sprite*.
34-49	Presentación.
50-61	Nivel 1.
62-70	Nivel 2.
71-79	Nivel 3.
80-88	Nivel 4.
89-98	Nivel 5.
99-110	Nivel 6.
111-120	Nivel 7.
121-128	Nivel 8.
129-140	Nivel 9.
141-155	Nivel 10.
156-158	Rutina de fallo.
159-172	Rutina de vidas y regreso a la pantalla.
173-181	Rutina de fin juego.
182-187	Control aleatorio de la pantalla izq., der.
188-193	Rutina ganador.
194-230	Generador de caracteres.
231-235	Animación del fallo del *sprite*.
236-241	Colores de la presentación.
242	Asigna récord.
243-245	Control mediante el *joystik* del *sprite*.
246-253	Efectos de color y música.

CONCURSO
PREMIADO CON
5.000
PESETAS

```
0 REM COPYRIGHT OSC AND ATQ 281086
1 MN$="OSC":VP=21000:V=53248:BA=49152
2 POKE53265,PEEK(53265)AND239
3 FORI=0TO142:READA:POKE49152+I,A:NEXT
4 POKE53280,0:POKE53281,0:POKEV+21,0
5 DATA169,0,133,251,169,4,133,252
6 DATA169,0,133,254,160,0,177,251
7 DATA133,253,160,1,177,251,136,145
8 DATA251,200,200,192,40,208,245,160
9 DATA39,165,253,145,251,165,251,105
10 DATA39,133,251,144,2,230,252,230
11 DATA254,166,254,224,25,208,213,96
12 DATA169,19,32,210,255,169,17,32
13 DATA210,255,169,157,32,210,255,169
14 DATA148,32,210,255,169,160,133,218
15 DATA169,13,32,210,255,96,169,0
16 DATA133,251,169,4,133,252,169,0
17 DATA133,254,160,39,177,251,133,253
18 DATA160,38,177,251,200,145,251,136
19 DATA136,192,255,208,245,160,0,165
20 DATA253,145,251,165,251,105,39,133
21 DATA251,144,2,230,252,230,254,166
22 DATA254,224,25,208,213,96,255
23 FORT=12288TO12350:READQ:POKET,Q:NEXT
24 DATA0,170,0,2,150,128,10,105,160
25 DATA9,170,96,39,170,152,26,151,228
26 DATA42,111,248,169,175,250,166,175
27 DATA250,154,159,246,106,107,233,169
28 DATA170,106,166,170,154,154,150,166
29 DATA106,105,170,41,170,104,38,170
30 DATA152,26,150,164,10,105,160,1,170
31 DATA64,0,170,0
32 GOSUB197
33 POKE53270,200:POKE53265,PEEK(53265)OR16
34 POKEV+21,0:PRINT"LOOPING BAL COPYRIGHT O S C":VI=3
35 PRINT"PULSA FUEGO"
36 PRINT"DESIGNADO"
37 PRINT"POR"
38 PRINT"DORIC BENAVENT
39 PRINT"RECORD: ";MN$;" ";VP
40 IFPEEK(56320)=111THEN42
41 GOSUB236:GOTO40
42 PRINT"INSTRUCCIONES"
43 PRINT"DEBES LLEVAR TU BOLA POR LOS LABERINTOS
44 PRINT"SIEMPRE TOCANDO LA RAYA,CUANDO NO LA TO-
45 PRINT"QUES PERDERAS UNA VIDA,TIENES 3 VIDAS,
46 PRINT"PERO NO TE FIES DEMASIADO EN EL NIVEL 3,
47 PRINT"VIGILA LOS CAMBIOS DE SENTIDO,CONECTA    EL JOYSTIK EN PORT 2
48 PRINT"PULSA UNA TECLA PARA CONTINUAR
49 GETA$:IFA$=""THEN49
50 POKEV+21,7:POKE53270,200
51 POKE53270,1:POKE2040,192:POKE2041,192:POKE2042,192:POKEV+40,2:POKEV+41,2
52 PA=1:POKEV+28,7:POKEV+37,5:POKEV+38,1:POKEV+39,2:POKEV+0,165
53 POKEV+1,180:F=160
54 GOSUB246
55 PRINT"..................................
56 POKEV+1,160:POKEV+2,255:POKEV+3,50:POKEV+4,225:POKEV+5,50:PU=1
57 GOSUB243
58 IFTI$="000020"THEN62
59 PU=PU+1:FO=5
60 IFPEEK(53279)=0THEN156
61 SYS49152:GOTO57
```

```
62 FO=5:PA=2:GOSUB246
63 POKEV+0,165
64 POKEV+1,160:F=160
65 PRINT"..▪.▪.┐.┐.┐.▪.▪.▪.┐.┐.┐.┐.┐.┐.▪.▪.▪.▪...■
66 POKEV+1,160:POKEV+2,255:POKEV+3,50:POKEV+4,225:POKEV+5,50
67 IFTI$="000020"THENGOTO71
68 PU=PU+1:GOSUB243
69 IFPEEK(53279)=0THEN156
70 SYS49238:GOTO67
71 PA=3:GOSUB246
72 POKEV+0,165
73 POKEV+1,160:F=160
74 PRINT"..▪.▪.┐.┐.┐.▪.▪.▪.┐.┐.┐.┐.┐.┐.▪.▪.▪.▪...■
75 FORA=0TO10:GOSUB243:FO=5
76 IFTI$="000020"THEN80
77 PU=PU+1
78 IFPEEK(53279)=0THEN156
79 FORX=0TO5:NEXT:SYSBA:NEXT:GOSUB182:GOTO75
80 PA=4:GOSUB246
81 POKEV+0,165
82 POKEV+1,160:F=160
83 PRINT"▪▪▪▪▪▪▪▪▪▪┐┐┐┐┐┐.┐┐┐...▪▪▪▪▪┐┐┐┐┐.......
84 IFTI$="000020"THEN89
85 FO=7:GOSUB243
86 PU=PU+1
87 IFPEEK(53279)=0THEN156
88 SYS49152:GOTO84
89 PA=5:GOSUB246:FO=7
90 POKEV+0,120:BA=49152
91 POKEV+1,150:F=150
92 PRINT"..┐┐┐┐┐┐▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪┐┐┐┐┐┐┐┐........
93 IFTI$>="000010"THENBA=49238
94 IFTI$="000020"THEN99
95 PU=PU+1:GOSUB243
96 IFPEEK(53279)=0THEN156
97 SYSBA
98 GOTO93
99 PA=6:GOSUB246
100 POKEV+0,155:BA=49152
101 POKEV+1,160:F=165:FO=8
102 PRINT"..┐┐┐┐┐┐▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪┐┐┐┐┐┐┐┐┐.......
103 GOSUB243
104 IFTI$>="000005"THENBA=49238
105 IFTI$>="000010"THENBA=49152
106 IFTI$>="000015"THENBA=49238
107 IFTI$="000020"THEN111
108 PU=PU+1
109 IFPEEK(53279)=0THEN156
110 SYSBA:GOTO103
111 PA=7:GOSUB246
112 POKEV+0,165:FO=8
113 POKEV+1,160:F=160
114 PRINT".....▪.┐.┐.┐.▪.▪.▪.┐.┐.┐.┐.┐.┐.▪.▪.▪.▪..■
115 PRINT".....▪.┐.┐.┐.▪.▪.▪.┐.┐.┐.┐.┐.┐.▪.▪.▪.▪..■
116 FORA=0TO12
117 IFTI$="000025"THEN121
118 PU=PU+1:GOSUB243
119 IFPEEK(53279)=0THEN156
120 SYSBA:NEXT:GOSUB182:GOTO116
121 PA=8:GOSUB246
122 POKEV+0,165
123 POKEV+1,160:F=160:FO=8
```

PROGRAMAS

```
124 PRINT"..■.■.┐.┐.┐.■.■.■.┐.┐.┐.┐.┐.┐.■.■.■.■...■
125 FORK=0TO20:GOSUB243
126 PU=PU+1:IFTI$="000020"THEN129
127 IFPEEK(53279)=0THEN156
128 SYSBA:NEXT:GOSUB182:GOTO125
129 PA=9:GOSUB246
130 POKEV+0,165
131 POKEV+1,160:F=160
132 PRINT"...■.■.┐.┐.┐.■.■.■.┐.┐.┐.┐.┐.┐.■.■.■.■..■
133 PRINT"■
134 PRINT"..■.■.┐.┐.┐.■.■.■.┐.┐.┐.┐.┐.┐.■.■.■.■..■
135 PRINT"■■"
136 FORK=0TO30
137 IFTI$="000020"THEN141
138 PU=PU+1:GOSUB243
139 IFPEEK(53279)=0THEN156
140 SYSBA:NEXT:GOSUB182:GOTO136
141 PA=10:GOSUB246
142 POKEV+0,165
143 POKEV+1,160:F=160
144 PRINT"■■■■■"
145 PRINT"■"
146 PRINT"...■ ■.┐.┐.┐.■.■ ■ ┐ ┐ ┐.┐.┐.┐ ■ ■ ■ ■  ■
147 PRINT"■"
148 PRINT". ■.■ ┐.┐ ┐.■ ■.■.┐.┐.┐.┐.┐ ┐.■.■.■ ■  ■
149 PRINT"  ■ ■.┐ ┐ ┐.■ ■ ■.┐.┐.┐ ┐.┐ ┐ ■.■ ■.■. ■
150 PRINT"■■"
151 FORK=0TO2
152 IFTI$="000025"THEN188
153 PU=PU+1:GOSUB243
154 IFPEEK(53279)=0THEN156
155 SYSBA:NEXT:GOSUB182:GOTO151
156 GOSUB231:FORL=54272TO54296:POKEL,0:NEXT
157 POKE54296,15:POKE54277,90:POKE54276,33
158 FORA=100TO0STEP-2:POKE54272,A:POKE54273,A:NEXT
159 VI=VI-1
160 IFVI=2THENPOKEV+21,3
161 IFVI=1THENPOKEV+21,1
162 IFVI=0THENPOKEV+21,0:GOTO174
163 IFPA=1THEN52
164 IFPA=2THEN62
165 IFPA=3THEN71
166 IFPA=4THEN80
167 IFPA=5THEN89
168 IFPA=6THEN99
169 IFPA=7THEN111
170 IFPA=8THEN121
171 IFPA=9THEN129
172 IFPA=10THEN141
173 FORA=0TO1000:NEXT
174 PRINT"[control characters]GAME OVER"
175 FORL=54272TO54296:POKEL,0:NEXT:POKE54278,190:POKE54277,90:POKE54278,190
176 POKE54276,33:POKE54296,15:FORA=155TO0STEP-5:POKE54272,A:POKE54273,A:NEXT
177 LETH=PU*100:FORA=0TOHSTEP+1000
178 PRINT"[control characters]PUNTUACIOON ";A:NEXTA
179 FORA=0TO1000:NEXT
180 IFH>=VPTHEN242
181 FORL=0TO1000:NEXT:GOTO34
182 MO=INT(4*RND(4))
```

```
183 IFMO=1THENBA=49152
184 IFMO=2THENBA=49238
185 IFMO=0THENBA=49238
186 IFMO=3THENBA=49152
187 RETURN
188 POKEV+21,0:PRINT"[cursor codes]OKEY"
189 PRINT"[cursor code] PERO SEGURO QUE NOS VEREMOS LAS CARAS
190 PRINT"[cursor code]     DE NUEVO EN LOOPING BAL II":FORA=0TO1500:NEXT
191 LETH=PU*100:FORA=0TOHSTEP+1000
192 PRINT"[cursor codes]PUNTUACIOON ";A:NEXTA
193 GOTO242
194 FORL=54272TO54296:POKEL,0:NEXT:POKE54278,190:POKE54277,90:POKE54278,190
195 POKE54276,33:POKE54296,15
196 FORA=0TO10:FORB=100TO0STEP-1:POKE54272,B:POKE54273,B:NEXT:NEXT
197 AA=14336:BB=53248:CC=56334
198 POKECC,PEEK(CC)AND254:POKE1,PEEK(1)AND251:FORJ=0TO2047:POKEAA+J,PEEK(BB+J):
199 NEXTJ:POKE1,PEEK(1)OR4:POKECC,PEEK(CC)OR1:POKE53272,31
200 FORJ=14336TO14551:READQ:POKEJ,Q:NEXT
201 FORJ=14704TO14711:READQ:POKEJ,Q:NEXT
202 DATA126,129,157,161,161,157,129,126
203 DATA30,33,33,63,33,33,33,0
204 DATA60,34,34,60,34,33,62,0
205 DATA31,32,32,32,32,32,31,0
206 DATA60,34,33,33,33,34,60,0
207 DATA63,32,32,60,32,32,63,0
208 DATA63,32,32,60,32,32,32,0
209 DATA63,32,32,39,33,33,63,0
210 DATA33,33,33,63,33,33,33,0
211 DATA4,4,4,4,4,4,4,0
212 DATA1,1,1,1,33,33,30,0
213 DATA33,34,36,56,36,34,33,0
214 DATA32,32,32,32,32,32,63,0
215 DATA33,51,45,33,33,33,33,0
216 DATA33,49,41,37,35,33,33,0
217 DATA63,33,33,33,33,33,63,0
218 DATA63,33,33,33,63,32,32,0
219 DATA63,33,33,33,33,57,63,3
220 DATA63,33,33,63,36,34,33,0
221 DATA63,32,32,63,1,1,63,0
222 DATA63,4,4,4,4,4,4,0
223 DATA33,33,33,33,33,33,63,0
224 DATA33,33,33,18,18,18,12,0
225 DATA65,65,65,65,34,42,20,0
226 DATA33,33,18,12,12,18,33,0
227 DATA33,33,18,18,12,4,4,0
228 DATA63,1,2,4,8,16,63,0
229 DATA126,129,165,189,189,153,129,126
230 RETURN
231 POKEV+23,1:GOSUB235:POKEV+29,1:GOSUB235:POKEV+23,0:GOSUB235:POKEV+29,0
232 A=0
233 A=A+1:POKEV+39,A:IFA>15THENPOKEV+39,2:RETURN
234 GOTO233
235 FORAS=0TO50:NEXT:RETURN
236 AD=INT(RND(1)*15)
237 FORA=55701TO55728:POKEA,AD:NEXT
238 FORB=55789TO55800:POKEB,AD+1:NEXT
239 FORC=55869TO55878:POKEC,AD:NEXT
240 FORD=55950TO55955:POKED,AD+1:NEXT
241 FORE=56105TO56132:POKEE,AD:NEXT:RETURN
```

```
242 POKE53270,200:INPUT"ENTRA TUS INICIALES";MN$:VP=H:GOTO34
243 IFPEEK(56320)=125THENF=F+F0:POKEV+1,F:RETURN
244 IFPEEK(56320)=126THENF=F-F0:POKEV+1,F:RETURN
245 RETURN
246 PRINT"NIVEL ";PA
247 A=1:FORZ=0TO15:FORDA=56232TO56248:GOSUB252:POKEDA,A:NEXT:NEXTZ
248 PRINT"":TI$="000000"
249 FORL=54272TO54296:POKEL,0:NEXT:POKE54278,190:POKE54277,90:POKE54278,190
250 POKE54276,33:POKE54296,15:FORK=0TO70STEP+5:POKE54272,K:POKE54273,K:NEXT
251 FORK=70TO0STEP-5:POKE54272,K:POKE54273,K:NEXT:RETURN
252 A=A+1:IFA>=15THENA=1
253 RETURN

READY.
```

CODIGOS DE CONTROL PARA EL VIC-20 Y EL C-64

Cómo se ve	Cómo se teclea	Efecto conseguido
Colores del VIC-20 y del 64		
	Ctrl + 1	Negro
	Ctrl + 2	Blanco
	Ctrl + 3	Rojo
	Ctrl + 4	Cian
	Ctrl + 5	Púrpura
	Ctrl + 6	Verde
	Ctrl + 7	Azul
	Ctrl + 8	Amarillo
Colores del 64 solamente		
	Cbm + 1	Naranja
	Cbm + 2	Marrón
	Cbm + 3	Rosa
	Cbm + 4	Gris oscuro
	Cbm + 5	Gris medio
	Cbm + 6	Verde claro
	Cbm + 7	Azul claro
	Cbm + 8	Gris claro

Cómo se ve	Cómo se teclea	Efecto conseguido
Códigos de cursor y control		
	Home	Cursor a casa
	Shift + home	Limpia pantalla
	Crsr	Cursor derecha
	Shift + crsr	Cursor izquierda
	Crsr	Cursor abajo
	Shift + crsr	Cursor arriba
	Ctrl + 9	Carácter inverso
	Ctrl + 0	Carácter normal
	Del	Borrar
	Shift + del	Insertar
Teclas de función		
	F1	
	F2 = Shift + F1	
	F3	
	F4 = Shift + F3	
	F5	
	F6 = Shift + F5	
	F7	
	F8 = Shift + F7	

commodore Magazine SERVICIO



Núm. 5
Programas, juegos y concurso/Londres: Quinta feria de Commodore/Basic, versión 4.75.



Núm. 6
El misterio del Basic/Lápices ópticos para todos/Concurso, juegos, aplicaciones.



Núm. 7
El ordenador virtuoso. MusiCalc. Programa monitor para el 64. Lápices ópticos. Ampliación de memoria para Vic-20.



Núm. 8
Joystick y Paddle para todos. Misterio del BASIC. EL LOGO. Cálculo financiero. Programas.



Núm. 9
Conversión de programas del Vic-20 al C-64. Móntale un paddle. Identifica tus errores. Software comentado.



Núm. 10
Koala Pad: La potencia de un paquete gráfico. Trucos. El FORTH. Software comentado. EL LOGO.



Núm. 11
Music-64. Supervivencia (1.ª parte). Cómo guarda el diskette la información. Sintetizador-64. El Forth (1.ª parte).

Núm. 12
Commodore-16 por dentro y por fuera. Sprites: los alegres duendecillos (1.ª parte). Supervivencia (1.ª parte). El Forth (y 3.ª parte).



Núm. 13
Análisis: programas de ajedrez. Los Cazafantasmas, 64. Vic en el espacio. La impresora que dibuja. Interface paralelo.



Núm. 14
Sprites; cómo entenderse con los duendes. Pilot: un lenguaje de alto nivel. Guía de Software para C-64.



Núm. 15
Síntesis de voz: su ordenador tiene la palabra. Pilot: un lenguaje de alto nivel (2.ª parte). Guía de software para C-64 (2.ª parte).



Núm. 16
Análisis de simuladores: vuela con tu C-64. Contabilidad para pequeños negocios. Cómo acelerar la ejecución de gráficos en BASIC. Submarino Commander. Pilot un lenguaje de alto nivel (3.ª parte).



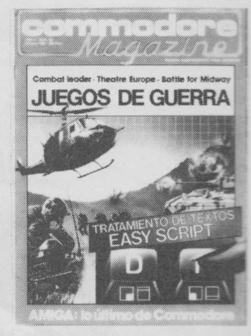
Núm. 17
Una lección de anatomía: los microordenadores por dentro. Bruce Lee: la furia oriental en el C-64. Quick Data Drive. Colossus Chess: un coloso del ajedrez.



Núm. 18
Practicalc: todo el poder de una hoja electrónica. Pascal (1.ª parte). Programas: juegos y aplicaciones veraniegos. ¡Canasta!: dos ases del baloncesto para el C-64.



Núm. 19
Análisis de cuatro impresoras. Simulación: pequeños mundos en su ordenador. Pascal (2.ª parte). Entombedy The Staff of Karnath: aventuras gráficas y mucha acción.



Núm. 20
Juegos de Guerra: Combat leader, Theatre Europe, Battle for Midway. Tratamiento de textos Easy Script. Amiga: lo último de Commodore. Libros, juegos y aplicaciones.



Núm. 21
Video-Digitizer: visión artificial para Commodore. Seikosha GP 700 VC: una impresora a todo color y con capacidades gráficas. Sprites multicolores. El nuevo C-128.

DE EJEMPLARES ATRASADOS

Complete su colección de COMMODORE MAGAZINE

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.



Núm. 22
Programas lightning: gráficos profesionales a tu alcance. Montaje: un interruptor programable para el C-64. Sprites múltiples. Cómo graba los datos el Datassette.



Núm. 25
Los ports: conexiones al exterior. El BASIC del C-128 Matemáticas por ordenador. Software educativo. Melodías musicales del Solfeo al Basic.



Núm. 27
Inteligencia Artificial: los lenguajes expertos. Sistemas de numeración: binario y hexadecimal. Comandos de disco C-128. Ficheros secuenciales. Libros, juegos y trucos.



Núm. 29
Piratas: problemas y legislación entorno a este tema. Backups: comentario del cartucho «Freeze Frame» y del Interface «Cosmos' Thoug». Introducción al PC de Commodore. Los juegos de Lucas Film.



Núm. 32
Gráficos en el Vic-20, C-64 y C-128. Hablemos de ordenadores o de cómo conseguir que no nos entienda nadie. Compatibles: fenómeno importante en el mundo de la microinformática profesional.



Núm. 23
Sinfonías en Chip: síntesis de sonido. Sprites en movimiento. Paisajes fractales en tu Commodore. Código máquina. Sight and sound: cuatro maestros de la música. Galería de Software: los mejores programas analizados en profundidad.



Núm. 26
Joysticks: ¿cómo son?, ¿cuáles son?, ¿cómo se programan?. Gráficos en código máquina. Simulador Spectrum. Proyecto Atenea.



Núm. 28
Robótica: evolución de la robótica y las posibilidades técnicas de los Robots. MECOMO: el brazo Robot. Mapa de memoria del C-128. Nuevas profesiones.



Núm. 31
La Magia del Amiga: primeras impresiones y posibilidades que este micro puede ofrecernos. Los periféricos que vienen. Mutaciones de personalidad.



Núm. 33
Reportaje fotográfico sobre la nueva imagen del C-64. Ofimática: la utilidad de los ordenadores en la empresa. Ordenadores de segunda mando. Juegos de Ingenio.



Núm. 24
El ordenador en la enseñanza. ADA para Commodore. El C-128 en fotos. Los peques y el ordenador. Seikosha Sp-1000 VC. Investigación sobre el sonido.

CORTE Y ENVIE ESTE CUPON A **COMMODORE MAGAZINE**

SERVICIO DE EJEMPLARES ATRASADOS

Bravo Murillo, 377. Tel.: 7337969 - 28020 MADRID

Ruego me envíen al precio de **300** ptas. los siguientes ejemplares atrasados de COMMODORE MAGAZINE ____________

El importe lo abonaré

Contra reembolso ☐ Cheque adjunto ☐ Con mi tarjeta de crédito ☐ American Express

☐ Visa ☐ Interbank

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Fecha de caducidad ____________

NOMBRE ____________

DIRECCION ____________

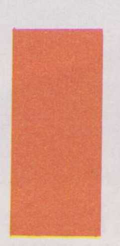
POBLACION ____________ C.P. ______

PROVINCIA ____________

A finales de los setenta, Dennis Ritchie diseñó en los laboratorios BELL un lenguaje de programación llamado **C.** Sus ideas principales hay que buscarlas en el BCPL, un lenguaje pionero en la técnica de programación estructurada (y del cual existe actualmente una versión para el QL). Del BCPL nació el B, y en este último se inspiró el **C.**

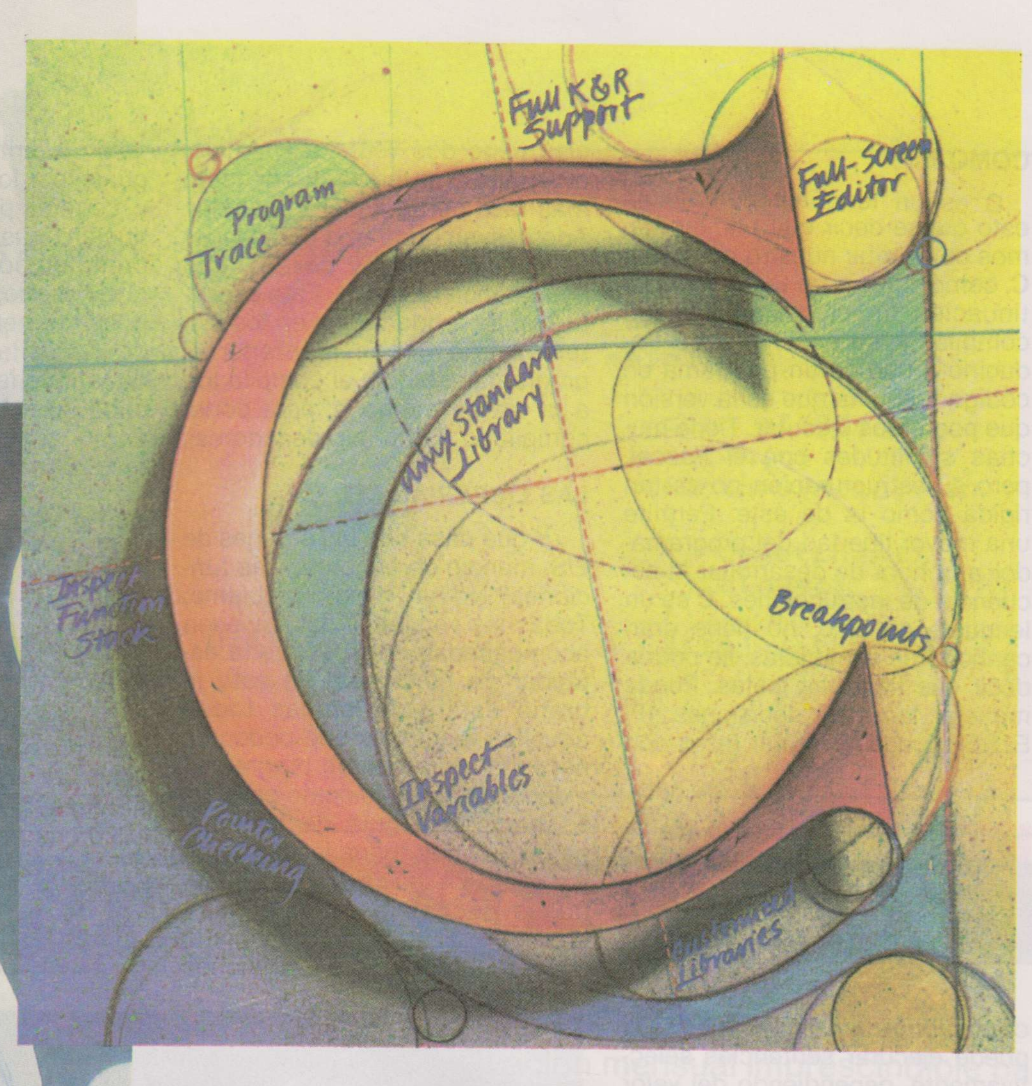
C Y UNIX

C se diseñó para un sistema operativo llamado UNIX, uno de los más potentes y usados hoy día. Un sistema operativo no es más que una colección de programas que controlan al ordenador y lo preparan para que podamos utilizarlo. Estos programas esta-



— El C ofrece una estructura libre que permite al programador trabajar «a su aire».

ban y están en las versiones más recientes de UNIX, escritos en **C.** Debido a esta aplicación, se piensa del **C** que es un lenguaje muy especializado y orientado hacia la escritura de estos complicados sistemas operativos. Sin embargo, la realidad es otra, el **C** es un lenguaje de objetivo general, es decir, igualmente sirve para realizar programas extremadamente simples, que se utiliza en la creación y desarrollo de complicados paquetes (programas combinados de uso específico). Con **C** se realizan infinidad de programas, desde tratamientos de textos, a programas de cálculos númericos, cubriendo entre ambos un amplio campo de utilidades. De hecho, ésta es la principal cualidad del **C,** su gran versatilidad, es un lenguaje todo-terreno.

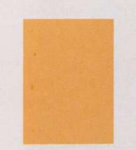


Cómo es y cómo funciona el más versátil y potente de los últimos lenguajes de programación.

Introducción al C

COMO ES

C es un lenguaje compilado, esto quiere decir que primero hemos de escribir nuestro programa **C**, como si fuera un texto, y a continuación un programa llamado compilador se encargará de traducirlo y darnos un programa en código máquina que es la versión que podremos ejecutar. Tiene muchas similitudes con el Pascal, pero su estructuración no es tan rígida como la de éste. Permite una mayor libertad del programador a la hora de desarrollar la secuencia de instrucciones. **C** es un lenguaje conciso, no tiene gran cantidad de sentencias, se podría decir que tiene las justas. Puede manejar toma de decisiones (IF-ELSE), puede controlar lazos con condiciones (FOR, WHILE, DO), puede ejecutar sentencias selectivamente, dependiendo del valor de una variable (SWITCH), y poco más. Trabaja con diversos tipos de variables que han de ser declarados, para cada variable que vayamos a utilizar hay que especificar su tipo (enteras, caracteres, doble precisión, etc.).



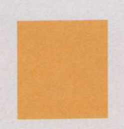
— El método de programación en C es tan simple como en BASIC.

¿Y ESTO ES TODO?

Por lo visto hasta ahora, no parece que el **C** sea nada del otro mundo. Además, hay que decir que tampoco tiene sentencias propias de entrada y salida, ni de manejo de ficheros, ni ninguna función (p. ej. no tiene OPEN#, CLOSE#, CAT#, SIN, EXP, CHR$, etc.). Ahora parece estar peor que antes, pero no es así. Todas estas características que parecen defectos, en realidad son virtudes, como vamos a ver. El tener un número pequeño de sentencias tiene dos ventajas, primero, el lenguaje es breve, y por tanto más fácil de aprender y manejar. Además, su programa compilador es muy reducido y por lo tanto ocupa poca memoria. Este aspecto resulta fundamental en los sistemas que deben mantener el programa escrito y el compilador, a la vez en memoria, para poder compilarlo (los miniordenadores).

LAS LIBRERIAS DEL C

¿Y qué pasa con las órdenes de E/S, manejo de ficheros y las funciones? No hay ningún problema. Todas las versiones del **C** vienen acompañadas de una librería estándar de funciones. En esta librería están contenidas todas aquellas funciones que podemos necesitar en nuestros programas y que el **C** no tiene como propias. Es como una guía de teléfono, nadie aprende todos los números de teléfonos de la ciudad; cuando es necesario saber alguno, simplemente va a la guía y allí lo consulta. Todas estas funciones están escritas en el propio lenguaje, por lo tanto se pueden leer y corregir para que funcionen a nuestro gusto, son como rutinas creadas por otro programador que nosotros queremos adaptar a nuestras necesidades. Así, cuando necesitamos alguna función de la librería, simplemente se escribe su nombre en el programa, como si fuese una instrucción más. El compilador se encargará de incluir todo el texto de la función en nuestro programa, para después compilarlo todo junto. Además, el **C** no sólo dispone de la librería estándar, también puede acceder a las librerías de funciones de otros lenguajes.



— El C es un lenguaje «todo-terreno».

Por lo visto hasta este momento, parece haber un empate entre



PROGRAMAS EJEMPLO

Veamos cómo sería, en BASIC y en C, un programa que acepte por teclado una cadena de sólo 10 caracteres.

BASIC:

```
10 DIM A$(10)
20 FOR N=1 TO 10
30 GET Z$ :IF Z$<>"" THEN GOTO 30
40 GET Z$ :IF Z$="" THEN GOTO 40
50 A$(N)=Z$
60 NEXT N
```

C:

```
main()
{
   char a[10];
   int n,c;
   for(n=0;n<10;++n)
   {
      c=getchar();
      a[n]=c;
   }
   a[10]='\0';
}
```

CUPON de SUSCRIPCION



Publicación mensual imprescindible para los usuarios de ordenadores personales **COMMODORE**

SOLO PARA ESPAÑA

Recorte este cupón debidamente cumplimentado, introdúzcalo en un sobre y envíelo a:



BRAVO MURILLO, 377-5.° A
28020 MADRID

Deseo recibir directamente en mi domicilio, **COMMODORE MAGAZINE,** durante un año (12 números), al precio de **2.700** ptas., lo que me supone un ahorro de **900** ptas.

NOMBRE ____________________

DIRECCION ____________________

CIUDAD ____________________ C.P. __________

PROVINCIA ____________________

Forma de pago: ☐ Talón
☐ Contrareembolso
☐ TARJETA DE CREDITO

Visa ☐ American Express ☐ Interbank ☐

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Fecha de caducidad ____________________

Firma,

Ahora un programa que calcula el factorial de un número. Se puede apreciar que el C y el PASCAL tienen muchas similitudes. El programa en C puede utilizarse también como una función para otros programas.

PASCAL:

```
program factorial (input,output);

var factorial : integer;
              : integer;

b
      i(n);
      orial := 1;
    while n > 0 do
       begin
          factorial := factorial * n;
          n         := n - 1
       end;
    write('el factorial de', n : 5, 'es: ', factorial)
end.
```

C:

```
factorial(n)
int n;
{
   int c;
   int factorial;
   factorial = 1;
   for(c=2;c<n+1;++c)
   {
      factorial=factorial*c;
   }
   printf("el factorial de %10d es: %10d \n", n, factorial);
}
```

Ahora veamos la creación de la función parte entera de forma artesana. El ejemplo en C puede ser almacenado en una librería, para utilizarse en otros programas que requieran esa función.

BASIC:

```
10 INPUT A
20 IF A>0 THEN GOTO 60
30 FOR N=0 TO A-1 STEP-1
40 IF A>N AND A<=N+1 THEN ENT=N+1 :GOTO 90
50 NEXT N
60 FOR N=0 TO A+1
70 IF A>=N AND A<N+1 THEN ENT=N :GOTO 90
80 NEXT N
85 STOP
90 PRINT ENT
100 GOTO10
```

C:

```
ent(n)
float n;
{
  int c;
  c=0;
  if(n<0)
  {
    while(c>n && n<c+1)
       --c;
  }
  else
  {
    while(c<n && n>c+1)
       ++c;
  }
  return(c)
}
```

el **C** y los demás lenguajes más conocidos. Hacen más o menos las mismas cosas y funcionan más o menos igual. Para romper el empate vamos a compararlos con el **C.** El COBOL y el FORTRAN son dos lenguajes fuertemente especializados. Surgieron de problemas muy concretos (la gestión de datos comerciales y el trabajo con complicadas fórmulas numéricas), ambos tienen unas estructuras muy rígidas, es decir, deben utilizarse en una forma muy concreta. Ambos tienen una cantidad enorme de comandos, uno para cada posible necesidad de cada campo de aplicación. Frente a ellos el **C** ofrece una estructura libre que permite al programador trabajar «a su aire». Es mucho más fácil de aprender, por su corto número de sentencias. Y ofrece todas las funciones que tienen estos lenguajes, pues como vimos tiene acceso a sus librerías. Así el **C** hace todo lo que estos lenguajes de una forma más simple.

FRENTE AL BASIC

Los anteriores eran lenguajes mono-propósito, con un único campo de aplicación. Veamos cómo es el lenguaje multi-propósito por excelencia, el BASIC. Puede realizar cálculos matemáticos y puede manejar información alfanumérica. Puede controlar ficheros y puede dibujar gráficas, parece ser el lenguaje universal. Pero su capacidad de cálculo y manejo de información tienen un límite. Además suele ser un lenguaje interpretado, el programa escrito reside en memoria y para ejecutarlo, el procesador va traduciendo cada sentencia del programa, como un intérprete de idiomas, según van llegando a ella. Esto ocasiona un problema de memoria que limita la extensión de los programas que pueden realizarse. Por el contrario el **C** con su pequeño tamaño permite realizar largos programas. Dispone de las funciones necesarias para el objetivo del programa, sacadas de su librería o de las librerías de otros lenguajes. Además el método de programación es tan simple como en BASIC, y estructurada. Pero presenta una ventaja adicional, surgida del mayor defecto del BASIC. **C** sólo hay uno, con más o menos funciones en sus librerías, pero sólo uno. Un programa **C** funciona igual en todos los ordenadores que tengan un compilador de **C.** Sin embargo, del BASIC hay mil y una versiones diferentes, casi tantas como modelos de ordenadores hay. Así, hay que escribir nuevas versiones de nuestros programas, si nos decidimos a cambiar de ordenador.





— C es breve, pocas sentencias que aprender y manejar.

CONCLUSION

Comparar el **C** con el PASCAL no resolverá ninguna duda. Realmente son dos lenguajes casi idénticos, tienen el mismo aspecto y funcionan de forma parecida. El PASCAL parece un hermano mayor del **C,** más serio, más rígido, más estructural, de mayor nivel. El **C** como hermano menor es más libre, más corto, menos estructurado, más cerca de la máquina. Son similares, ambos hijos de una misma madre, la última o penúltima generación de lenguajes de programación, por eso sus cualidades son parecidas. Resumiendo, el **C** es un lenguaje corto, con pocas sentencias que aprender y un compilador que ocupa poca memoria. Al ser tan corto no es nada especializado, lo cual le permite trabajar en cualquier campo, verdaderamente multipropósito. Con una estructura no rígida, fácil de manejar. Con unas posibilidades muy amplias gracias a sus librerías de funciones. Y es completamente estándar, un programa **C** funciona en cualquier ordenador que soporte el lenguaje.

EXITO DEL C

Un lenguaje puede ser potente pero arduo de utilizar (p. ej. ENSAMBLADOR). Puede ser específico, pero exigir una gran disciplina a la hora de escribir los programas (p. ej. COBOL). Y sin embargo estos dos lenguajes son ampliamente utilizados, a pesar de sus defectos. El éxito de un lenguaje no se mide en su potencia, ni en su idoneidad para un trabajo. Lenguajes existen muchos, lenguajes aceptados universalmente, muy pocos. ¿Cuál es la clave del éxito? Esa clave se encuentra en el programador; si se siente cómodo con un lenguaje, ese lenguaje ha triunfado. El **C** es uno de esos lenguajes. El programador se siente muy cómodo trabajando en **C.** Puede hacer de él lo que quiera, hay total libertad. Desde usarlo como un BASIC o un PASCAL (con sus funciones), hasta hacer de él un semi-ensamblador.

RITO Y BAJO NIVEL

El **C** sólo exige definir los tipos de las variables que vamos a utilizar en nuestro programa. A partir de ahí todo es libre. Sólo tenemos que escribir nuestro programa. Podemos ir llamando a las funciones que necesitamos, el compilador se encargará de montarlas en nuestro programa. Ahora estamos trabajando lejos de la máquina, en alto nivel. También podemos, al crear nuestro programa, ir creando nuestras propias funciones. Crear nuestras entradas y salidas, las peticiones de datos, las funciones de cadenas de caracteres, y las matemáticas. Así creamos nuestra librería de funciones, que podemos ir ampliando y utilizarla en otros programas. Estamos trabajando cerca de la máquina, haciéndola bailar a nuestro aire, trabajamos en bajo nivel.

A NUESTRO AIRE

Se puede personalizar el lenguaje. Incluso crear un nuevo sublenguaje que nos guste más, todo ello a base de crear rutinas de funciones, que serán los comandos del nuevo dialecto. Esta es la gran cualidad del **C,** su libertad. Podemos llegar a dominar la máquina, sin demasiadas complicaciones, y con pocas instrucciones que aprender. El **C** ofrece mucho y no pide demasiado. Si te gusta programar, te encantará.



— Un programa C funciona en todos los ordenadores que tengan un compilador C.

Joaquín Mateos Lago

BIBLIOGRAFIA

— **El lenguaje de programación C**
Kernighan y Ritchie. Prentice Hall. 1978.

— **Learning to program in C**
Thomas Plum. Prentice Hall. 1983.

Las ocho damas y los autómatas que se reproducen



Ocho damas en un tablero de ajedrez, sin poderse comer unas a otras. Probablemente, frente a este problema, que propusimos la semana pasada, algunos habrán mirado un tablero de ajedrez, y tras unos pocos e infructuosos intentos habrán desistido y habrán considerado como imposible la solución, o, cuando menos, harto complicada. Quizás, otros más tenaces hayan encontrado la solución a mano, poniendo una y otra vez las damas hasta encontrarla. Y, posiblemente, algunos pocos hayan intentado y conseguido hacer un programa de ordenador que pudiera resolverlo. La sorpresa habrá sido grande al notar que pocas cosas son tan sencillas de programar en BASIC como la solución a este problema. En poco más de 15 líneas podemos hacer que el Commodore dé una o todas las soluciones posibles al problema. ¿Cómo? Empleando una vez más la recursividad. Al igual que antes, cuando resolvíamos el laberinto, cada paso lo suponíamos válido hasta que o bien encontrábamos la solución o bien encontrábamos el camino cortado, en cuyo caso deberíamos volver para atrás, y probar soluciones diferentes, en lo que llamábamos *backtracking*. Este problema es exactamente el mismo. Cada «paso» es en este caos una reina colocada, y el algoritmo que usamos es entonces el de suponer que una dama está correctamente colocada y probar entonces la siguiente.

La primera reina la colocaremos en la primera columna y en cualquier fila, digamos por ejemplo, la primera. A partir de aquí, veo si es atacada por alguien (que evidentemente en el caso de ser la primera dama es imposible), y si no lo es, pruebo otra nueva reina en la columna siguiente, en este caso las dos. Nuevamente volveríamos a ha-

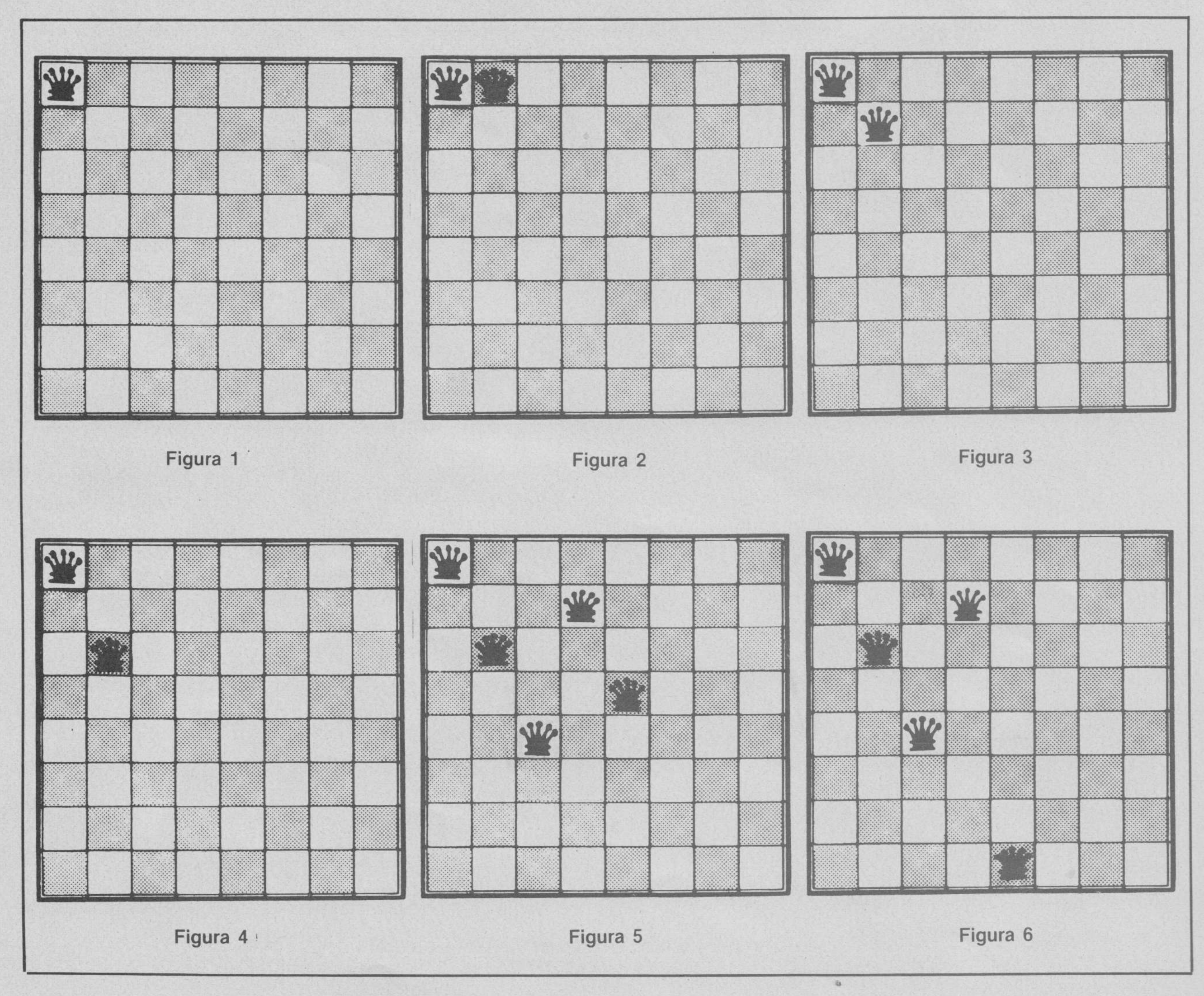

Figura 1

Figura 2

Figura 3

Figura 4

Figura 5

Figura 6

cer lo mismo, comprobaría hasta encontrar una fila donde la dama no es atacada y posteriormente se llamaría otra vez a sí misma intentando colocar otra dama en una nueva columna. ¿Y si no es posible poner la dama en ninguna fila sin que sea atacada? Pues eso significa que alguna de las damas anteriores está mal puesta, de forma que volvemos a la dama anterior y la movemos de sitio hasta encontrar una nueva posición donde no sea atacada y podamos continuar hacia adelante. Vamos a ver cómo funciona con un ejemplo gráfico.

La figura 1 tiene la primera dama colocada en la primera fila y en la primera columna. La segunda dama empezaríamos colocándola tal como muestra la figura 2.

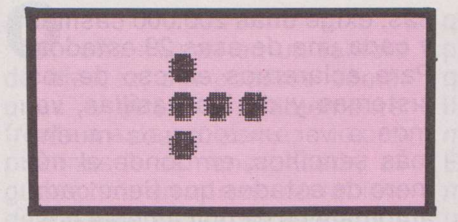
Pero no es válida ya que está frente a la primera. Seguimos moviendo la dama como muestra la figura 3 hasta que en la figura 4 encontramos una posible posición. Volvemos a hacer lo mismo con la siguiente dama hasta que llegamos a la figura 5, en que no existe ninguna fila, dentro de la columna 6, que sea una posición válida. Aplicamos entonces el *backtracking,* y volvemos hacia atrás con la dama anterior, y la colocamos tal y como muestra la figura 6. Si tampoco es válido, iríamos hacia atrás hasta encontrar una dama con la que podamos continuar adelante. El proceso acaba cuando conseguimos llegar a la octava fila y colocar la dama en una posición aceptable. El programa que aparece listado hace exactamente eso, y en la instrucción número 20 es donde coloca la primera dama. Si queréis que esa primera dama esté en una posición distinta de la primera fila, sólo tenéis que modificar esa instrucción. Cada caso tiene un tiempo de búsqueda distinto, y si queréis sacar todas las soluciones que existen, cambiar el STOP de la línea 500 por k=k-1:RETURN.

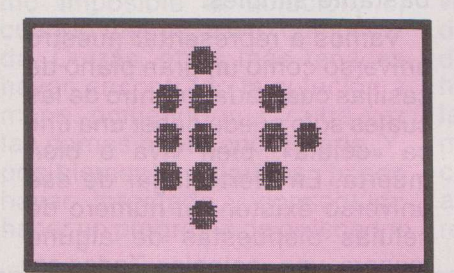
Para aquellos que consideren interesante este tipo de problemas, pueden intentar este otro también muy clásico. Consiste en recorrer, desde la casilla de la primera fila y la primera columna, todas y cada una de las 64 casillas que componen un tablero de ajedrez, con un caballo, pasando una sola vez por cada sitio. Se basa en todo lo que hemos comentado anteriormente, y es casi más fácil que el problema de las ocho damas.

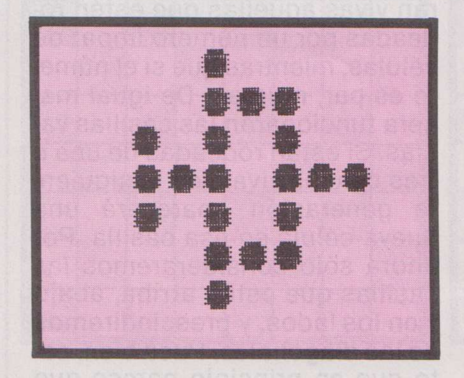
LOS AUTOMATAS QUE SE REPRODUCEN

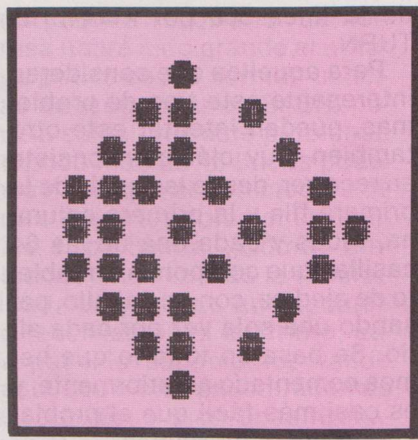
Vamos a hacer un alto momentáneo en nuestra carrera por la consecución de la inteligencia artificial y vamos a tocar un tema que no por menos en boga, resulta menos interesante que el anterior: se trata de la reproducción de los ordenadores. Dicho así, suena realmente a utópico el pensar que un ordenador pueda reproducirse, pero la realidad es que en los primeros años, en que se desarrollaron los ordenadores y las actuales teorías de la computación, uno de los temas que primero se tocó fue precisamente el de la posibilidad de que un ordenador reprodujera su propia configuración. Los primeros estudios de esos sistemas los iniciaron en la década de los cincuenta John Von Neumann (probablemente el padre de los ordenadores actuales) y Stanislaw Ulam, e intentaron recoger una serie de reglas para establecer cómo debería comportarse lo que se llamó autómatas celulares. Von Neumann pudo demostrar que sí existen configuraciones autoreproductivas, demostrando que existe un constructor universal capaz de crear cualquier configuración, y capaz, por consiguiente, de crear la correspondiente a sí mismo. Pero nosotros no vamos a estudiar ese sistema, ya que, entre otras cosas, exige unas 200.000 casillas y cada una de esas 29 estados. Para aclararnos en eso de los sistemas y de las casillas, vamos a ver unos casos mucho más sencillos, en donde el número de estados que tiene cada autómata es de únicamente dos, pongamos de 0 y 1 (a modo de *bits),* y donde las reglas serán bastante simples.

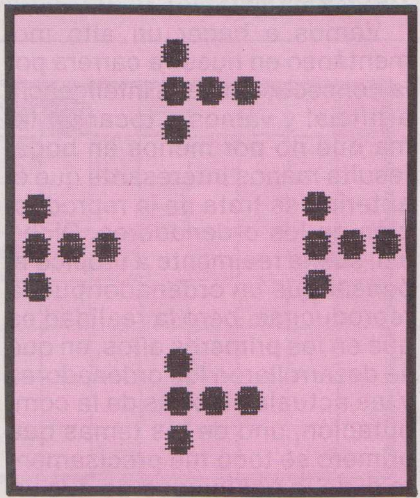
Vamos a representar nuestro universo como un gran plano de casillas cuadradas, dentro de las cuales sólo puede haber una única «célula», bien viva o bien muerta. En cierto lugar de ese universo existen un número de células dispuestas de alguna manera, una «colonia». Todas estas células viven un cierto período de tiempo, pero en la generación siguiente, sólo permanecerán vivas aquellas que estén rodeadas por un número impar de células, mientras que si el número es par, mueren. De igual manera funcionarán las casillas vacías. Si están rodeadas de una o tres células vivas, en la siguiente generación aparecerá una nueva célula en esa casilla. Por ahora sólo consideraremos las casillas que están arriba, abajo y en los lados, y prescindiremos de las diagonales. Pues bien, esto que en principio parece que generará colonias más o menos de formas aleatorias, e incluso se podría pensar que acabarían desapareciendo, resulta que con estas reglas, todas las colonias quedan repartidas, exactamente igual, pasadas una serie de generaciones. La más simple de todas las colonias es una única célula. En la generación siguiente, las cuatro células que están ortogonales a ella aparecerán vivas, con lo cual la colonia origen se repite desde la primera generación. Pero podéis pensar en colonias todo lo complicadas que queráis, que al final vuelven a aparecer, y además cuatro de ellas dispuestas simétricamente respecto a los ejes. Pensad en


1.ª Generación


2.ª Generación


3.ª Generación


4.ª Generación


5.ª Generación

esa colonia de células como si fuera un ser formado por **n** células, digamos un ser mecánico, con capacidad de reproducir su elemento mecánico más simple, la «célula», y que obedeciera a esas leyes de reproducción de sus elementos más simples. En un período de tiempo finito acabaría haciendo cuatro ejemplares exactamente igual que él.

No sólo en el campo de la ficción tienen este tipo de problemas aplicación, sino que muchas veces se pueden simular casos reales con este tipo de sistemas, cada uno con unas reglas diferentes. Un caso muy bonito lo forman las moléculas de vapor de agua y hielo que existen en la atmósfera. Cuando están juntos ambos elementos y existe posibilidad de que se forme nieve, cada copo va conformándose con unas reglas parecidas a las que hemos visto, y las «células» son, en este caso, moléculas de agua, y existen en este caso dos tipos de «celúlas» las de vapor de agua y las de hielo. Ambos tipos tienen unos enlaces de forma que podemos simularlas como celdillas de panal, ya que son hexagonales, como las figuras de un panal de abeja o los puntos de los balones de fútbol. Todas están rodeadas de una serie de moléculas de vapor y de hielo, y si el número de moléculas de hielo que las rodea es impar acaban helándose. Hecho esto por ordenador, y reproduciendo las moléculas de vapor en rojo y las de hielo en azul, podemos reproducir la creación de un copo de hielo a partir de una configuración que nosotros demos. Y aunque parezca mentira, el copo de hielo que se os funde en la mano cuando os cae en ella es de una belleza increíble a los ojos de un microscopio. Si lográis coger uno y que no se os licue, cuando le obervéis al microscopio, pensad que ese copo es único, como las obras de arte, ya que nunca hay dos iguales. Cada uno tiene la forma que le confiere la colonia que lo generó. Resulta muy curioso realizar esto con un ordenador, y en el listado de la figura 1 podéis comprobarlo vosotros mismos. En todas las figuras que vosotros introduzcáis las sucesivas generaciones que se producen van dando lugar a una floresta de células que adquieren las formas más variopintas, a modo de un caleidoscopio. Si tenéis suficiente imaginación, podéis incluso crear dos colonias, una de tiburones y otra de peces, y podéis ver cómo evolucionan los dos bancos de peces y tiburones, donde las reglas serán las que decidan el futuro de vuestro mar.

Sobre los «juegos de la vida», sin duda el más famoso es el de John Horton Conway, de la Universidad de Cambridge, presentado por Martin Gardner en 1970, en Scientific American.

En este caso, el objeto del juego no es que el sistema se autoreproduzca, sino generar la máxima cantidad de pautas, o de configuraciones posibles, para que sea imposible adivinar si

acabarán muriéndose o permanecerán como una floreciente especie en el mundo de los autómatas celulares. Las reglas son, en este caso, algo más difíciles de llevar a la programación, pero no exigen demasiado. En este caso, en lugar de tener en cuenta únicamente las células ortogonales, es decir, las de arriba, los lados etc., se consideran también las diagonales, lo que da lugar, muchas veces, a que aparezcan «ramas» que se extienden en la pantalla y bifurcaciones en esas direcciones diagonales. Las reglas son como siguen. Si una célula está viva, sólo podrá seguir viviendo si tiene dos o tres vecinas vivas. Si el número de células que le rodea es menor, la célula está aislada, no puede reproducirse y muere. Pero si el número es mayor, existe una superpoblación, y el alimento no es suficiente, con lo que muere irremediablemente. Cuando la célula está muerta, o, lo que es lo mismo, la casilla está vacía, sólo puede volver a la vida si tiene exactamente tres células vivas a su alrededor. También es repetitivo, pero no aparecen cada número fijo de generaciones. En la figura 2 está el programa que simula estos ejemplos, aunque es susceptible de mejora si se desea seguir realmente a la colonia durante un gran número de generaciones, ya que entonces deberemos realizarlo en alta resolución. Si sois capaces de llevar a la práctica todo un mundo de marcianos en algún lejano planeta donde cada tipo de marciano llevase un color, podréis crear unas figuras realmente artísticas, que se van alargando como hongos o que desaparecen después de una floreciente población. Basta con dejar volar la imaginación... y al Commodore.

Alejandro de Mora Losana

```
10 DIM V(9)
20 V(1)=1
30 K=2
100 IF K=9 THEN GOSUB 300 :END
110 FOR M=1 TO 8
120 GOSUB 200
130 IF BOOL=1 THEN V(K)=M :K=K+1 :GOSUB 100 :K=K-1 :M=V(K)
140 NEXT M
150 RETURN
200 BOOL=1
210 FOR I=1 TO K-1
220 IF V(I)=M OR ABS(V(I)-M)=ABS(K-I) THEN BOOL=0 :RETURN
230 NEXT I
240 RETURN
300 DIM A(8,8)
310 FOR I=1 TO 8 : A(J(I),I)=1 : NEXT I
320 FOR I=1 TO 8 : FOR J=1 TO 8
330 PRINT A(I,J);"-";
340 NEXT J : PRINT
350 NEXT I
360 RETURN
```

Este programa no funciona en tu Commodore, pues no dispone éste de suficiente pila de direcciones, para poder comprobar su funcionamiento puedes utilizar cualquier otro ordenador (Amstrad, Spectrum, MSX, etc...)

```
1 POKE53280,0:POKE53281,0
5 PRINT "[CLR]";
10 DIM A(24,39)
20 DIM B(24.39)
```

```
29 REM **** FIGURA EN MATRIZ A ****
30 A(12,19)=1
40 A(12,20)=1
50 A(12,21)=1
60 A(11,19)=1
70 A(13,19)=1
79 REM **** LIMITES A MIRAR F/C ***
80 NF=10 :AF=14
90 IC=18 :AC=22
99 REM ** PINTA 1EXP GENERACION ***
100 GOSUB 2000
199 REM * CALCULA GENERACION EN B *
500 FOR I=NF TO AF
510 FOR J=IC TO AC
520 SU=A(I-1,J)+A(I+1,J)+A(I,J-1)+A(I,J+1)
530 IF SU=1 OR SU=3 THEN B(I,J)=1
540 IF SU=2 OR SU=4 OR SU=0 THEN B(I,J)=0
550 NEXT J
560 NEXT I
599 REM ** PASA MATRIZ A HACIA B **
600 FOR I=NF TO AF
610 FOR J=IC TO AC
620 A(I,J)=B(I,J)
650 NEXT J
660 NEXT I
670 NF=NF-1 :AF=AF+1 :IC=IC-1 :AC=AC+1
699 REM **** AMPLIAMOS LIMITES ****
700 GOSUB 2000
799 REM **** NUEVA GENERACIONE ****
800 IF NF>0 AND AF<25 AND IC>0 AND AC<40 THEN GOTO 500
899 REM ** MIRA LIMITES PANTALLA **
1000 END
1999 REM *** PINTA GENERACIONES ***
2000 FOR N=NF TO AF
2010 FOR M=1 TO 39
2012 NUM=983+N*40+M:N2=NUM+54272
2015 IF A(N,M)=1 THEN POKE NUM,81:POKE N2,7
2020 IF A(N,M)=0 THEN POKE NUM,32:POKE N2,0
2030 NEXT M
2040 NEXT N
2050 RETURN
```

Programa número 1

```
1 POKE53280,0:POKE53281,0
5 PRINT "♥";
10 DIM A(24,39)
20 DIM B(24,39)
29 REM **** FIGURA EN MATRIZ A ****
```

```
30 A(11,19)=1
40 A(11,20)=1
50 A(11,21)=1
60 A(12,19)=1
61 A(12,20)=1
62 A(12,21)=1
70 A(13,19)=1
71 A(13,20)=1
72 A(13,21)=1
79 REM **** LIMITES A MIRAR F/C ***
80 NF=10 :AF=14
90 IC=18 :AC=22
99 REM ** PINTA 1EXP GENERACION ***
100 GOSUB 2000
199 REM * CALCULA GENERACION EN B *
500 FOR I=NF TO AF
510 FOR J=IC TO AC
520 S1=A(I-1,J)+A(I+1,J)+A(I,J-1)+A(I,J+1)
521 S2=A(I-1,J-1)+A(I-1,J+1)+A(I+1,J-1)+A(I+1,J+1)
522 SU=S1+S2
530 IF SU=2 OR SU=3 AND A(I,J)=0 THEN B(I,J)=1
531 IF SU=1 OR SU=4 AND A(I,J)=0 THEN B(I,J)=0
540 IF SU=3 AND A(I,J)=1 THEN B(I,J)=1
541 IF SU=1 OR SU=2 OR SU=4 AND A(I,J)=1 THEN B(I,J)=0
550 NEXT J
560 NEXT I
599 REM ** PASA MATRIZ A HACIA B **
600 FOR I=NF TO AF
610 FOR J=IC TO AC
620 A(I,J)=B(I,J)
650 NEXT J
660 NEXT I
670 NF=NF-1 :AF=AF+1 :IC=IC-1 :AC=AC+1
699 REM **** AMPLIAMOS LIMITES ****
700 GOSUB 2000
799 REM **** NUEVA GENERACIONE ****
800 IF NF>0 AND AF<25 AND IC>0 AND AC<40 THEN GOTO 500
899 REM ** MIRA LIMITES PANTALLA **
1000 END
1999 REM *** PINTA GENERACIONES ***
2000 FOR N=NF TO AF
2010 FOR M=1 TO 39
2012 NUM=983+N*40+M:N2=NUM+54272
2015 IF A(N,M)=1 THEN POKE NUM,81:POKE N2,7
2020 IF A(N,M)=0 THEN POKE NUM,32:POKE N2,0
2030 NEXT M
2040 NEXT N
2050 RETURN
```

Programa número 2

Podrá parecer una perogrullada para muchos de los lectores, pero, probablemente resulte operativo refrescar algún concepto sobre la eficacia del llamado «Octavo Bit», que tantos sinsabores suele causar a los usuarios finales de una configuración estándar de un ordenador y una impresora. Ese bit que puede provocar el que una impresora no funcione como nosotros quisiéramos.

Los secretos del Octavo Bit



Como es sabido, todos los ordenadores, sea cual sea su marca, para obtener una eficacia real en su acción técnica, suelen ir aunados a una impresora. Entre ambos existe un canal de comunicación, generalmente estandarizado a través de una salida en serie, RS232, una salida Paralelo-Centronics, u otras alternativas menos conocidas, que no vienen ahora al caso.

Deteniéndonos un momento, al observar los gráficos que acompañan a este artículo, podremos recordar que si se genera un texto en las zonas de memoria del ordenador, se producirá un especial almacenamiento de datos que serán los que el ordenador envíe a la impresora.

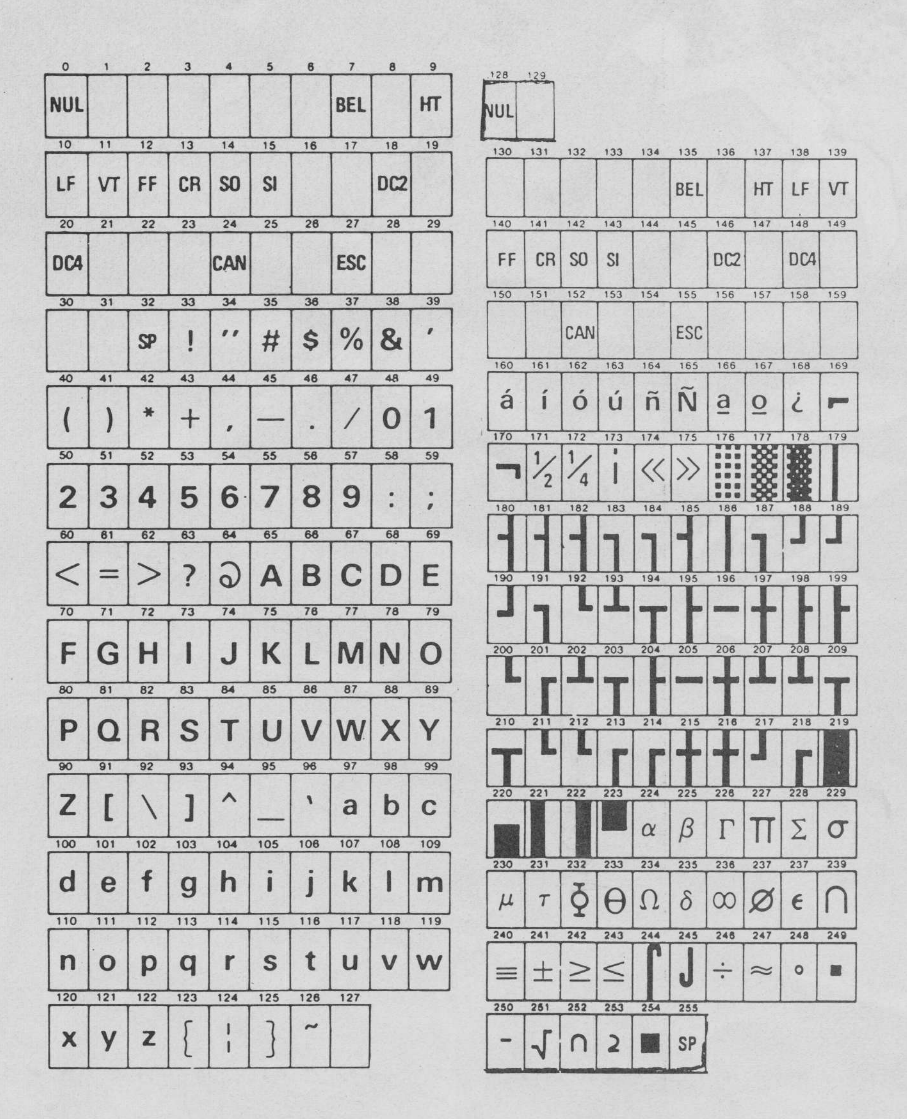
La unidad informática por excelencia es el *byte,* formada por 8 *bit,* mínimas determinaciones digitales binarias, que, operando bajo combinaciones de **0** y **1,** construyen los elementos constitutivos formales de cualquier representación binaria que comporte un estado lógico; susceptible de ser manifestado gráficamente, vía pantalla o impresora. En nuestro caso, la actividad del octavo *bit* la cifraremos en su versión a impresora, dada la importancia que, últimamente, han adquirido los programas y aplicaciones que exigen una importante resolución gráfica. Además, dependerá de que hayamos codificado bien ese *bit* para poderle sacar el mejor rendimiento a nuestras máquinas.

Como se ha aceptado generalmente, esas combinaciones de 0 y 1, en bloques de 8 dígitos (y bajo una estructura lógica predeterminada), constituyeron el código internacional ASCII **(American Standard Code for Information Interchage).** (Ver cuadro 1). A través de su uso, se podía transmitir, en forma binaria absolutamente estructurada, toda la información



	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø			SP	Ø	@	P	`	p			SP	Ø	@	P	`	p
1		DC1	!	1	A	Q	a	q		DC1	!	1	A	Q	a	q
2		DC2	"	2	B	R	b	r		DC2	"	2	B	R	b	r
3		DC3	#	3	C	S	c	s		DC3	#	3	C	S	c	s
4		DC4	$	4	D	T	d	t		DC4	$	4	D	T	d	t
5			%	5	E	U	e	u			%	5	E	U	e	u
6			&	6	F	V	f	v			&	6	F	V	f	v
7	BEL		'	7	G	W	g	w	BEL		'	7	G	W	g	w
8	BS	CAN	(	8	H	X	@	x	BS	CAN	(	8	H	X	h	x
9	HT		)	9	I	Y	i	y	HT		)	9	I	Y	i	y
A	LF		*	:	J	Z	j	z	LF		*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{	VT	ESC	+	;	K	[	k	{
C	FF		,	<	L	\	l	¦	FF		,	<	L	\	l	¦
D	CR		-	=	M	]	m	}	CR		-	=	M	]	m	}
E	SO		.	>	N	^	n	~	SO		.	>	N	^	n	~
F	SI		/	?	O	_	o		SI		/	?	O	_	o	

Tabla 1. Código ASCII.



0	1	2	3	4	5	6	7	8	9
NUL							BEL		HT
10	11	12	13	14	15	16	17	18	19
LF	VT	FF	CR	SO	SI			DC2	
20	21	22	23	24	25	26	27	28	29
DC4				CAN			ESC		
30	31	32	33	34	35	36	37	38	39
		SP	!	"	#	$	%	&	'
40	41	42	43	44	45	46	47	48	49
(	)	*	+	,	—	.	/	0	1
50	51	52	53	54	55	56	57	58	59
2	3	4	5	6	7	8	9	:	;
60	61	62	63	64	65	66	67	68	69
<	=	>	?	@	A	B	C	D	E
70	71	72	73	74	75	76	77	78	79
F	G	H	I	J	K	L	M	N	O
80	81	82	83	84	85	86	87	88	89
P	Q	R	S	T	U	V	W	X	Y
90	91	92	93	94	95	96	97	98	99
Z	[	\	]	^	_	`	a	b	c
100	101	102	103	104	105	106	107	108	109
d	e	f	g	h	i	j	k	l	m
110	111	112	113	114	115	116	117	118	119
n	o	p	q	r	s	t	u	v	w
120	121	122	123	124	125	126	127		
x	y	z	{	¦	}	~			

Cuadro 2.0. Caracteres con el 8.° bit a 0.

128	129								
NUL									
130	131	132	133	134	135	136	137	138	139
					BEL		HT	LF	VT
140	141	142	143	144	145	146	147	148	149
FF	CR	SO	SI			DC2		DC4	
150	151	152	153	154	155	156	157	158	159
		CAN			ESC				
160	161	162	163	164	165	166	167	168	169
á	í	ó	ú	ñ	Ñ	a̲	o̲	¿	⌐
170	171	172	173	174	175	176	177	178	179
¬	½	¼	¡	«	»	░	▒	▓	│
180	181	182	183	184	185	186	187	188	189
┤	╡	╢	╖	╕	╣	║	╗	╝	╜
190	191	192	193	194	195	196	197	198	199
╛	┐	└	┴	┬	├	─	┼	╞	╟
200	201	202	203	204	205	206	207	208	209
╚	╔	╩	╦	╠	═	╬	╧	╨	╤
210	211	212	213	214	215	216	217	218	219
╥	╙	╘	╒	╓	╫	╪	┘	┌	█
220	221	222	223	224	225	226	227	228	229
▄	▌	▐	▀	α	β	Γ	π	Σ	σ
230	231	232	233	234	235	236	237	237	239
µ	τ	Φ	Θ	Ω	δ	∞	Ø	ϵ	∩
240	241	242	243	244	245	246	247	248	249
≡	±	≥	≤	⌠	⌡	÷	≈	°	▪
250	251	252	253	254	255				
-	√	∩	2	■	SP				

Cuadro 2.1. Caracteres con el 8.° bit a 1.

que el ordenador remitiría a la impresora. En ciertos aspectos, resulta indiferente el hecho de que las impresoras sean matriciales (de agujas), de margarita o láser. Lo que sí hay que tener presente es que cada impresora tiene su CPU y sus memorias con una labor muy concreta, una tarea fija «de propósito dedicado», a saber reconocer y dar salida a los caracteres enviados, sean para producir simbolismos y signos gráficos, o bien, espacios en blanco, órdenes para la acción *carry return, line feed,* etc. En estos últimos casos, por abuso de lenguaje, se les ha considerado caracteres aunque, propiamente hablando no lo son, sino que codifican un movimiento, ejecutan un control de los movimientos de la impresora. Esto es muy importante en el tema que nos ocupa, ya que se certifica que existen informaciones «no visibles», caracteres «colaboradores», en la transmisión de datos desde el ordenador a la impresora.

Haciendo historia, en los inicios de la definición de los caracteres ASCII, se observó que no era absolutamente necesario que el ordenador enviase a la impresora los ocho *bits* de cada *byte:* bastaba con utilizar siete para poder representar todos los caracteres ASCII. A la hora de escribir en inglés, la estructura ASCII era operativa con sólo siete *bits,* de manera que el octavo sobraba: podía indicarse en él un 1 o un 0, indiferentemente. De hecho, así se hizo —dejando absoluta libertad de definición a ese octavo *bit*— hasta que empezaron a surgir problemas con la irrupción de culturas distintas a la norteamericana en el entorno informático.

Como es sabido, el inglés excluye de su simbología gráfica toda una serie de signos como son la letra **ñ,** la apertura de admiración y la de interrogación y los acentos, así como incluye en su repertorio un peculiar uso de las comillas. Asimismo, signos

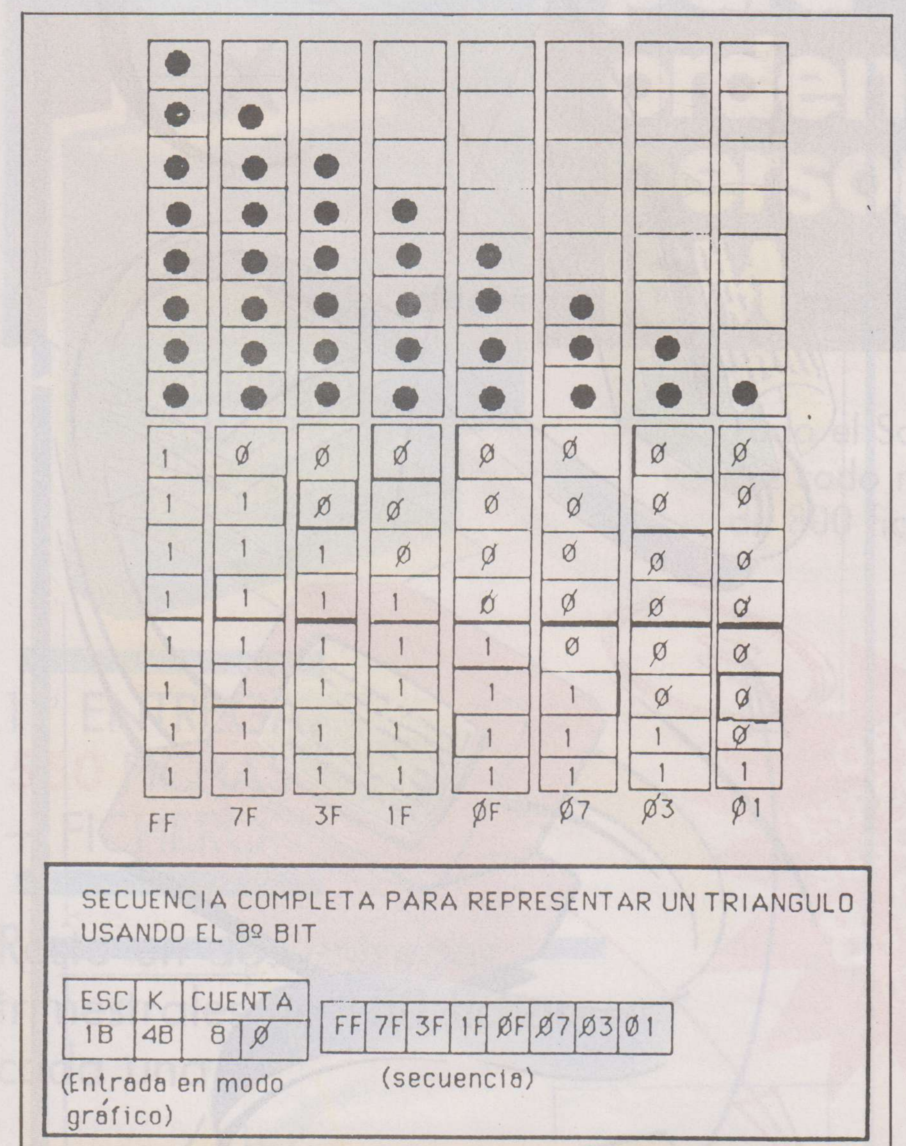


Tabla 3. Representación de un triángulo.

como ***, $, &,** entre otros, eran de escaso uso en países que no fuesen del área idiomática de influencia anglosajona.

En ese momento, fue cuando empezó a hacerse patente la eficacia del octavo *bit*. Respetando los siete que conforman la estructura internacional ASCII, pero, a la vez, dándole una función específica al *bit* libre, se podían variar los contenidos alfabéticos, numéricos o simbólicos propios del esquema preestablecido. Con la libre disposición de ese *bit* ciego, no operativo hasta que se le indique lo contrario, los fabricantes poseen una herramienta utilísima a la hora de plegar la relación ordenador-impresora a sus deseos; pero, al mismo tiempo —y como es harto frecuente— la utilización libre del octavo *bit* ha supuesto un paso más (tal vez, de los más importantes y definitorios) dentro de la «babelización», de la ceremonia de la confusión que se está dando en el mundo informático. La libre disposición de ese *bit* ha sido una de las fuentes más relevantes en el espinoso asunto de las incompatibilidades entre equipos de fabricantes distintos.

Volvamos a la historia. Los creadores del ASCII informaron al mundo de que había un *bit* libre que podía ser programado y activado a gusto del usuario. La proliferación de vocabularios informáticos, de simbologías locales, provocaría fenómenos tan definitivos como el que un usuario final que tuviese una unidad central de fabricación norteamericana, un teclado francés, una impresora japonesa y un *software* español estuviese en un permanente estado de «neurosis informática»... El problema se puede soslayar a base de realizar un correcto tratamiento de los *switches* de la impresora (generalmente, los manuales pueden ser bastante clarificadores al respecto), y una reprogramación que incluya la función deseada para ese octavo *bit,* de cara a su adecuación a los símbolos que ocupan el teclado en uso.

Una de las funciones operativas más importantes y definitorias del papel que juega el octavo *bit* activo, se da cuando se trabaja en modo gráfico.

En cualquier caso —y a despecho de lo enunciado en el título del artículo—, el famoso octavo *bit* es el menos secreto de todos los componentes de una familia *byte,* ya que se le utiliza, por activa o por pasiva, para crear cualquier tipo de construcción original que se salga de los tipos estandarizados del ASCII. En manos del fabricante y del usuario final está la última palabra.

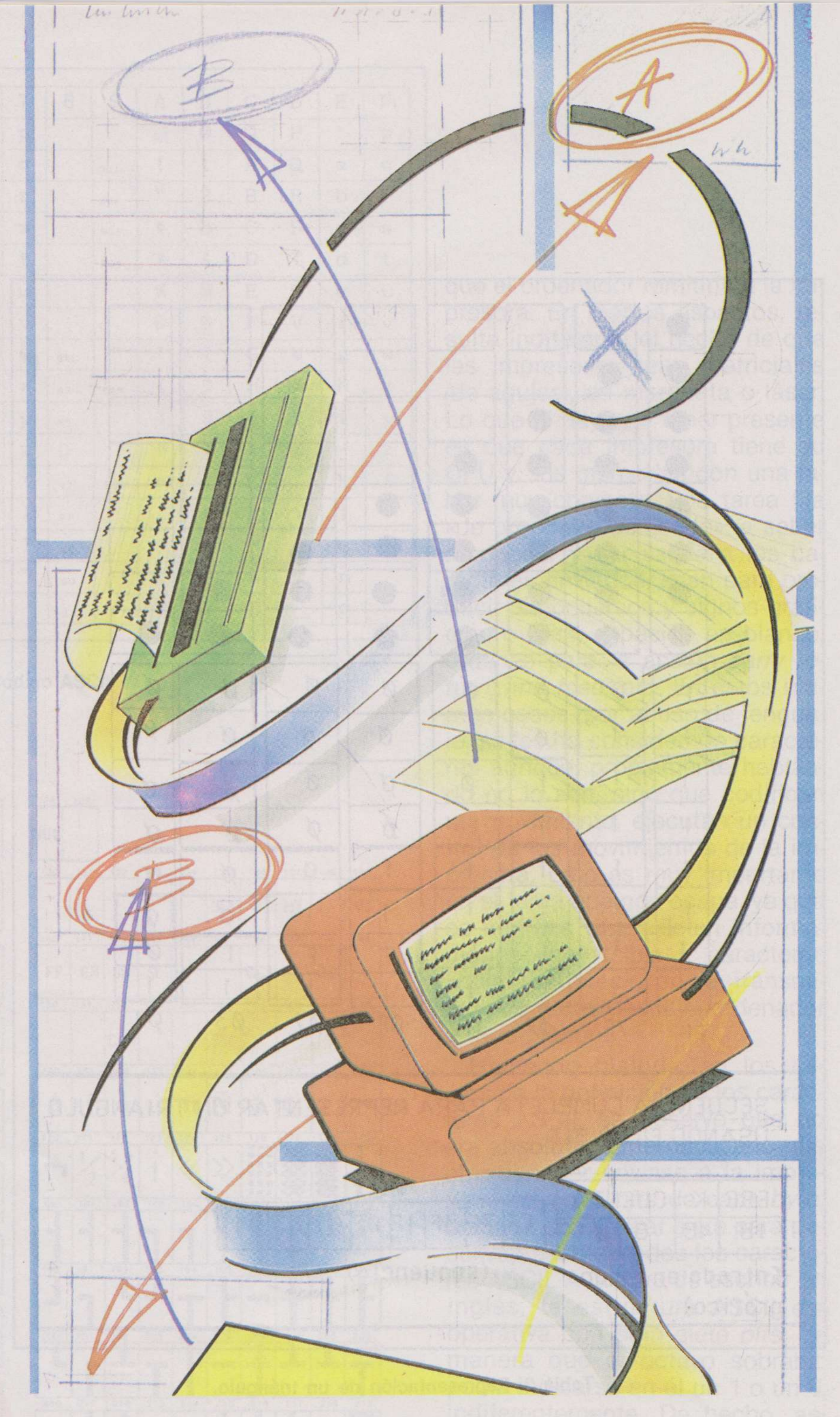
EJEMPLO: GENERACION DE UN TRIANGULO

Supongamos que el usuario final de una configuración estándar (ordenador + impresora) desea generar una figura geométri-

ca que no existe en el repertorio de caracteres que ofrece el ASCII (y que, por tanto, no es reconocido por su impresora y no la emite, o, como mucho, la imprime incompleta o defectuosamente). Elijamos un modelo cualquiera. Por ejemplo, un triángulo. Viendo el grafismo incorporado en la Tabla 3, podemos ver que está formado por ocho columnas y ocho filas, en orden decreciente (al observar el resultado en el papel de la impresora, con una sencilla lupa advertiríamos su estructura, la huella que cada pulso de las agujas han dejado sobre el papel).

Jugando con la ya citada orden «escape», indicando nuestro deseo de entrar en «modo gráfico», tenemos una posibilidad radical de utilizar la eficacia del octavo *bit*. Como de costumbre, el fabricante nos indicará cuál es la mecánica o proceso para acceder al modo gráfico. En este caso, habiendo utilizado una de las informaciones más vulgarizadas, tenemos que la orden de acceso al modo gráfico nos viene dada por las órdenes **1B** y **4B** (véase gráfico adjunto). Asimismo, deberemos precisar cuál es la longitud de ese grafismo que deseamos realizar —en este caso un triángulo—, para determinar el volumen de su presencia en el texto a emitir. Por ello, tendremos que indicar la cuenta de los *bytes* que deberán enviarse para cerrar la secuencia que hemos elegido. El triángulo escogido cuenta con ocho pulsaciones de impresora —valga la expresión— de altura y, a su vez, su base también está formada por ocho puntos. De ahí que la tercera orden que debamos remitir a la impresora sea 80 (conceptuada bajo el espacio «cuenta» en la tabla anexa al ejemplo).

Todo lo anterior equivale a decir que, durante ocho *bytes*, nuestra impresora va a estar trabajando en modo gráfico, pero que, transcurrida esta secuencia, volverá a su modo normal, recuperando su régimen habitual de actividad.



Nuestro deseo, en este ejemplo, era determinar la figura de un triángulo de ocho puntos de lado (aunque, geométricamente hablando, no resulte muy ortodoxa esta definición...). El primer paso, obviamente, sería determinar su altura. Si observamos el diseño adjunto, exige ocho pulsaciones, o, lo que es lo mismo, ocho certificaciones de *bits* activos. Con los siete utilizados en las configuraciones normales ASCII no nos sería suficiente. Hay que recurrir al octavo *bit*...

En el ejemplo que hemos adjuntado, siguiendo las normas que exige el correcto uso de las alternativas que el *bit* libre nos ofrece, se presenta la secuencia completa para representar esta figura geométrica, de acuerdo a nuestra voluntad, una vez que tengamos a la impresora en modo gráfico.



cartas

Rutinas en código máquina

P: Mi problema consiste en que no sé dónde tengo que colocar las rutinas en código máquina que salen en las revistas para que, a la hora de utilizarlas desde mis programas escritos en BASIC, no se pierdan y alteren el funcionamiento de éstos.

También quisiera saber cuánto espacio es posible tener en memoria reservado para las rutinas en código máquina y, si no es posible, qué tengo que hacer yo para reservar ese espacio.

Adolfo de Blas
Madrid

R: El mejor lugar para colocar las rutinas en código máquina en la memoria de tu Commodore se encuentra en la zona comprendida entre $C000 y $CFFF, suponiendo que éstas ocupen menos de 4 *KBytes.* Pues en ese caso sería necesario preparar una zona de memoria de forma que los programas Basic no tengan permitido el paso.

El final de la memoria reservada para los programas Basic se encuentra en $9FFF y nosotros podemos cambiarlo por una rutina de la KERNAL llamada MEMTOP o mediante unas instrucciones Basic.

Estas son las instrucciones.

```
POKE 51,L
POKE 52,H
POKE 55,L
POKE, 56,H
CLR
```

La H es el *Byte* alto y la L el *Byte* bajo de la nueva dirección del final de memoria.

Por ejemplo, si deseas reservar una dirección de memoria desde la $9000 hasta $9FFF. Para tus programas en lenguaje máquina tienes que utilizar las siguientes instrucciones.

```
POKE 51,0
POKE 52,144
POKE 55,0
POKE 56,144
CLR
```

SCROLL

P: ¡Hola! Os escribo con el propósito de que me respondáis a estas dos preguntas.

1) ¿Qué es el SCROLL?

2) ¿Cómo se puede hacer?

David Rincón
Barcelona

R: (1) Un SCROLL es un desplazamiento del contenido de la pantalla en cualquiera de las cuatro direcciones básicas (arriba, abajo, derecha e izquierda). Se utiliza muy frecuentemente en los juegos donde el escenario de acción es más grande que la pantalla.

(2) En los desplazamientos de pantalla por Scroll, normalmente lo primero que se hace es reducir la pantalla a 38 columnas o 25 líneas, según sea la dirección elegida. Esto se consigue POKEando las direcciones 53270 ó 53265. La reducción de pantalla es necesaria, pues en un Scroll, en cada desplazamiento de 8 bits, se inserta una nueva línea o columna que en el caso de estar reducida la pantalla no es apreciada por nosotros.

La explicación no es más profunda, pues consideramos, que para realizar un Scroll, es necesario que tengas unos conocimientos avanzados de programación que en pocas líneas no podemos explicar. Es posible que en poco tiempo escribamos un artículo sobre las posibilidades de Scroll en el Commodore, mientras tanto te aconsejo que busques información en algún libro que hable en profundidad de los gráficos con el Commodore, y sobre una base de conocimiento, nos plantees las dudas que te puedan surgir.

Programación

P: Me he dado cuenta de que en muchos de los programas que sacais en la revista hay subrutinas que no son necearías. ¿Por qué se hacen los programas de esta forma, pudiendo hacerse más cortos con sentencias GOTO?. ¿Es para entenderlos mejor?. Siento mucha curiosidad por saber por qué programais de esa manera.

Jorge García
Barcelona

R: La manera de programar depende del factor al que demos primacía: Rapidez, Comprensión, Ahorro de memoria, Fiabilidad, etcétera...

Cada día los ordenadores disponen de más memoria y de mayor velocidad de cálculo. Ganando en importancia la programación estructurada y de fácil comprensión, así como la fiabilidad de los programas.

Un programa bien construido, donde los errores sean fáciles de detectar, las modificaciones fáciles de introducir y que para cualquier modificación, el trabajo requerido sea mínimo, este programa podemos decir que mantiene una programación estructurada y profesional.

Llamamos programación estructurada a la creación de programas a base de bloques, donde cada bloque realiza una función determinada y usa a su vez unos bloques más pequeños, y así sucesivamente. Quedando al final un programa a base de subrutinas, lo más independientes y generales posibles, que es lo que tú puedes apreciar en los programas con sucesivas llamadas GOSUB.

La diferencia entre GOTO Y GOSUB es clara, a un GOSUB accedemos desde cualquier punto, siempre que queramos, encargandose el RETURN de recordar la dirección de vuelta, en cambio con GOTO, aparte de hacerse más difícil la comprensión del programa, sólo podemos utilizarlo una vez, debiendo direccionar la vuelta con otro GOTO.

DATAMON news

DATAMON — DATAMON, S. A. — REPRESENTACION EN ESPAÑA DE: RITEMAN — PROVENZA, 385-387 TEL. (93) 207 24 99* — TELEX 9 [illegible]791 08025 BARCELONA

A Vd. que ya nos conoce por las impresoras

RITEMAN

y confía en nosotros por la calidad, servicio y garantía



Gama F+/C+ — Gama R10 — Gama R15

le ofrecemos ahora también los ordenadores personales compatibles-asequibles



Peceman®

los más avanzados tecnológicamente y con la mejor relación precio-prestaciones



Gama 8088 (4,77 Mhz)



Gama Turbo (4,77 y 8 Mhz)



Gama AT Turbo (8 y 10 Mhz)

De venta en los mejores establecimientos especializados